EE206A - Spring 2005

Lecture #6: FOUNDATIONS OF SENSOR NETWORK DESIGN

Aman Kansal
UCLA - NESL
kansal@ee.ucla.edu
http://nesl.ee.ucla.edu

Reading List for this Lecture

- **Most Relevant**
  - J. Cortés, S. Martìnez, and F. Bullo. Analysis and design tools for distributed motion coordination. In American Control Conference, Portland, OR, June 2005. http://nesl.ee.ucla.edu/courses/ee206a/2005s/papers/L06/Cortes05_ACC.pdf
  - Chapter 1 of [Mackay 2003] http://www.inference.phy.cam.ac.uk/mackay/itila/book.html or, http://nesl.ee.ucla.edu/courses/ee206a/2005s/papers/L06/Mackay03_book.pdf
  - Mark Paskin, Carlos Guestrin and Jim McFadden, A Robust Architecture for Distributed Inference in Sensor Networks, In the Fourth International Conference on Information Processing in Sensor Networks (IPSN'05), April 2005. http://nesl.ee.ucla.edu/courses/ee206a/2005s/papers/L06/Paskin05_IPSN.pdf

- **Optional**
  - David Mackay, Information Theory, Inference, and Learning Algorithms, 2003. http://www.inference.phy.cam.ac.uk/mackay/itila/book.html or, http://nesl.ee.ucla.edu/courses/ee206a/2005s/papers/L06/Mackay03_book.pdf
  - Huiyu Luo and Gregory Pottie, 2005, Routing Explicit Side Information for Data Compression in Wireless Sensor Networks, IEEE DCOSS. http://nesl.ee.ucla.edu/courses/ee206a/2005s/papers/L06/Luo05_DCOSS.pdf
  - P. Gupta and P. R. Kumar, 2000, "The capacity of wireless networks," *IEEE Transactions on Information Theory* , vol. IT-46, no. 2, pp. 388-404, March 2000. http://nesl.ee.ucla.edu/courses/ee206a/2005s/papers/L06/Gupta00_ITIT.pdf
  - Bhardwaj, M., A.P. Chandrakasan, 2002, "Bounding the Lifetime of Sensor Networks Via Optimal Role Assignments", *INFOCOM 2002,* pp. 1587-1596, New York, June 2002, http://nesl.ee.ucla.edu/courses/ee206a/2005s/papers/L06/Bhardwaj02_Infocom.pdf
  - CONTD….

Reading List for this Lecture

Optional (contd.)

– J. Chen, X. Zhang, T. Berger and S. B. Wicker "The Sum–Rate Distortion Function and Optimal Rate Allocation for the Quadratic Gaussian CEO Problem" IEEE Journal on Selected Areas in Communications: Special Issue on Sensor Networks, Volume 22, No. 6, August 2004.

– A Kansal, A Ramamoorthy, M Srivastava, G Pottie, "On Sensor Network Lifetime and Data Distortion" IEEE International Symposium on Information Theory, 2005. http://nesl.ee.ucla.edu/courses/ee206a/2005s/papers/L06/Kansal05_ISIT.pdf

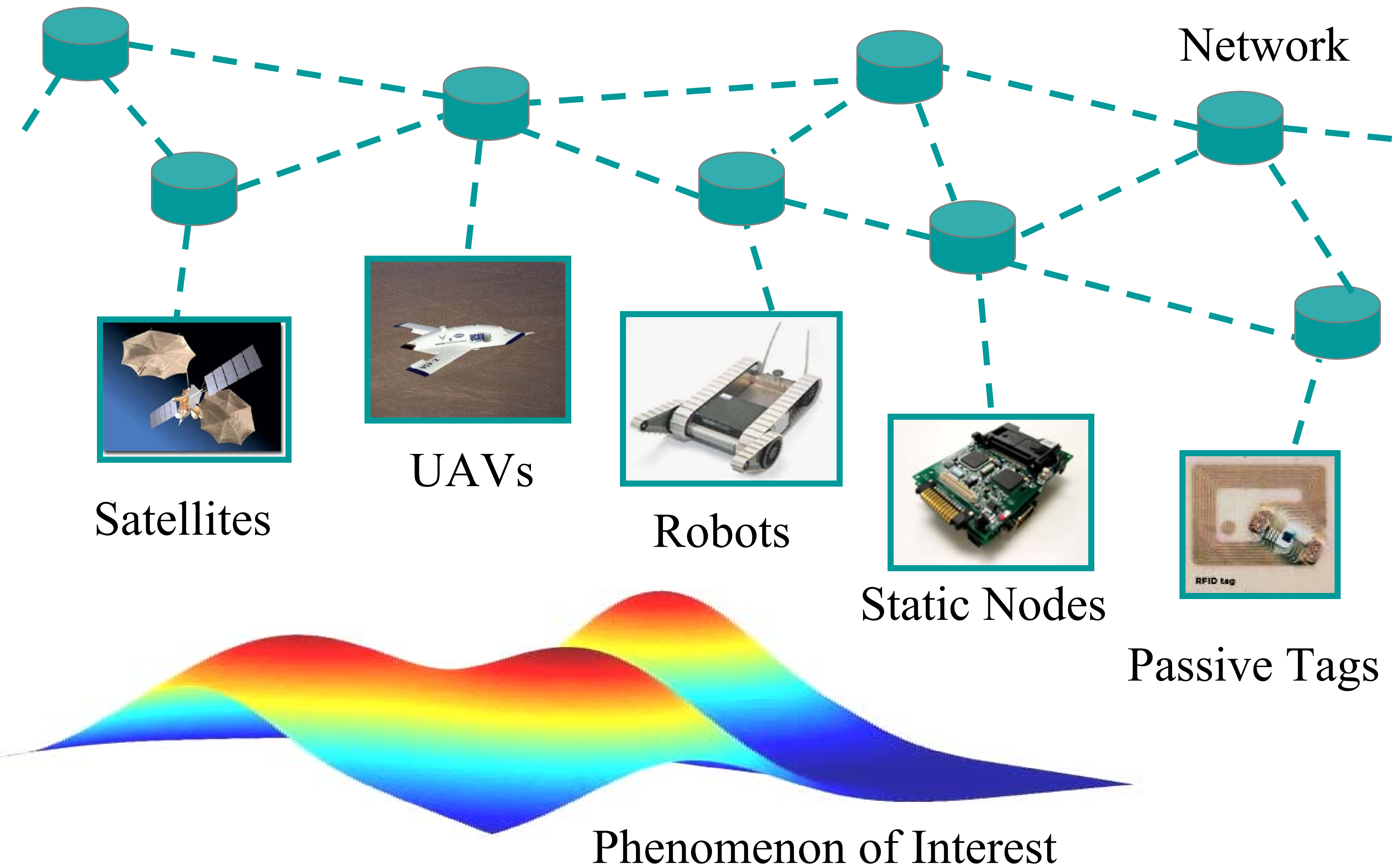
– TM Cover and J Thomas, "Elements of Information Theory", John Wiley, 1991. http://www-isl.stanford.edu/~jat/eit2/index.shtml

– B. Grocholsky. "Information-Theoretic Control of Multiple Sensor Platforms." The University of Sydney, Ph.D Thesis, 2002. http://nesl.ee.ucla.edu/courses/ee206a/2005s/papers/L06/Grocholsky02_USydney.pdf

– Arvind Giridhar and P. R. Kumar, ``Computing and communicating functions over sensor networks." IEEE Journal on Selected Areas in Communications. pp. 755--764, vol. 23, no. 4, April 2005. http://nesl.ee.ucla.edu/courses/ee206a/2005s/papers/L06/Giridhar05_JSAC.pdf

– Seapahn Meguerdichian et al , Exposure in wireless Ad-Hoc sensor networks, ACM Mobicom 2001. http://nesl.ee.ucla.edu/courses/ee206a/2005s/papers/L06/Meguerdichian01_MobiCom.pdf

The basic design problem



The basic design problem

- Design a network to extract desired information about a given discrete/continuous phenomenon with high fidelity, low energy and low system cost
- Questions:
  - Where should the data be collected
  - What data and how much data should be collected
  - How should the information be communicated
  - How should the information be processed
  - How can network configuration be adapted in situ
- Single tool does not answer all questions

Multiple Tools for Specific Instances

- Problem broken down into smaller parts
  - Coverage and deployment: Geometric optimization
  - Data fusion: Information theory, estimation theory, Bayesian methods
  - Communication: network information theory, geometric methods, Bernoulli random graph properties
  - Energy performance: Network flow analysis, linear programming
  - Configuration adaptation with mobile nodes: adaptive sampling theory, information gain

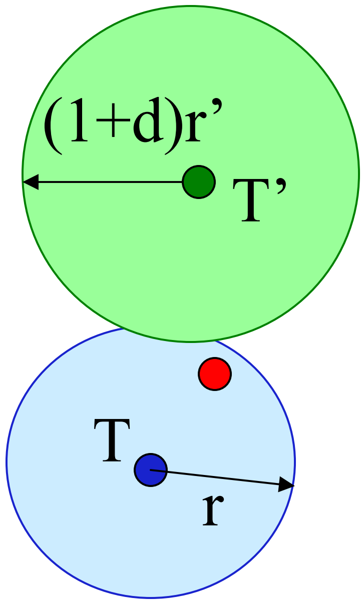
- Not a comprehensive list

Communication

Network Capacity and Node Density

[Gupta and Kumar 2000]

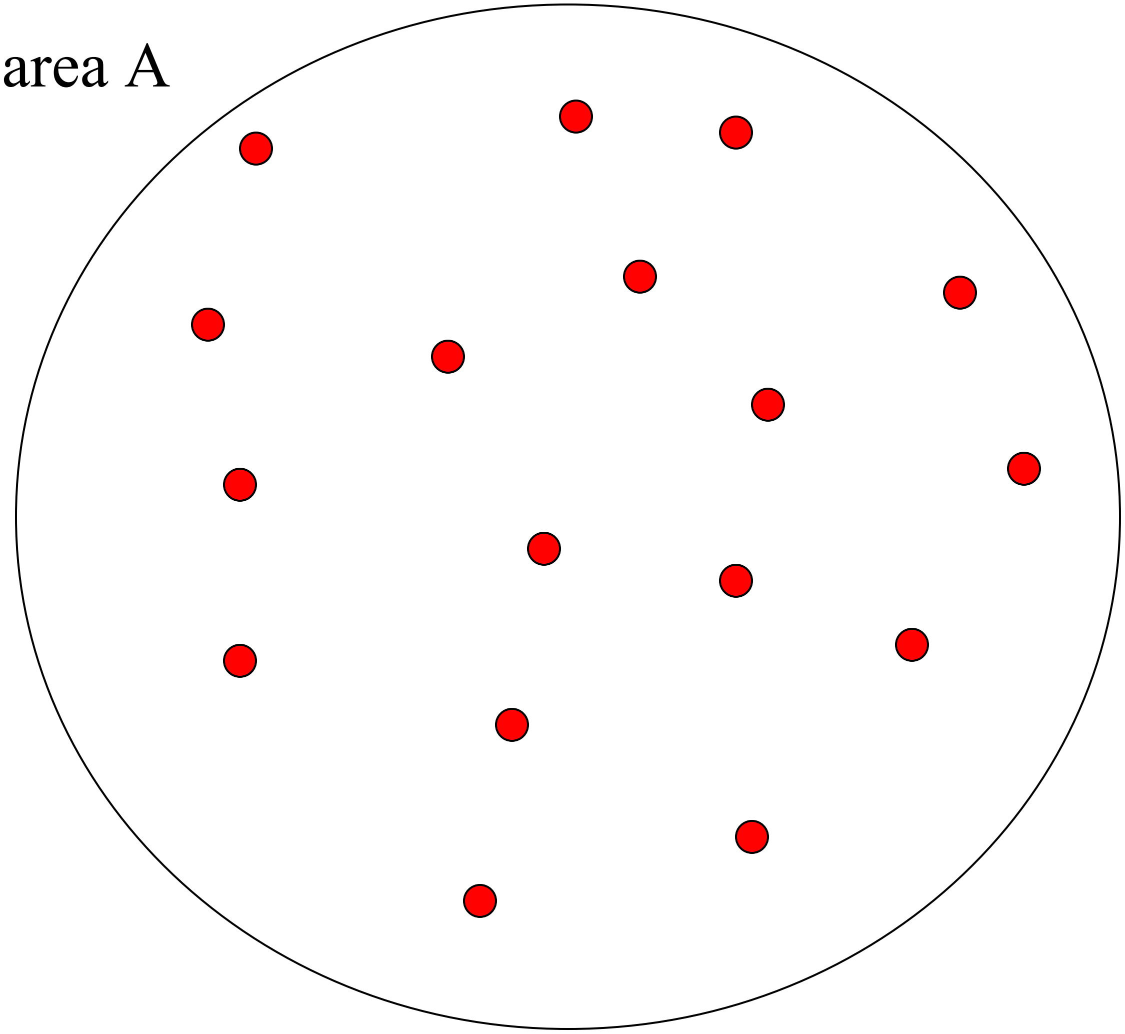
- A simple model for a wireless network:
  - Disk of area A
  - N nodes
  - Each node can transmit at W bps
- Transmission range: disk of radius r
  - r depends on transmit power
  - Interference range (1+d)r
- Communication is successful if
  - Receiver within r of transmitter
  - No other interfering transmission in (1+d)r’ of receiver



ACK: Slide adapted from PR Kumar’s plenary talk at Sensys 2004.

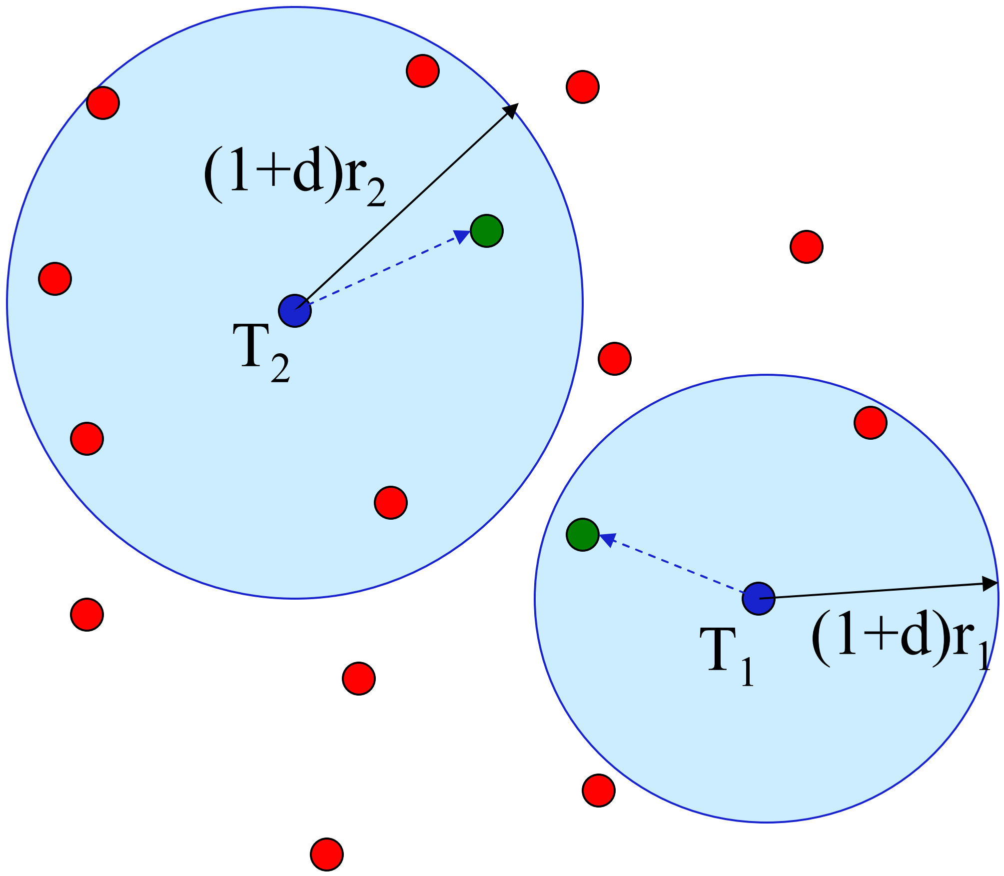
How many simultaneous transmissions?

N nodes in area A



ACK: Slide adapted from PR Kumar's plenary talk at Sensys 2004.

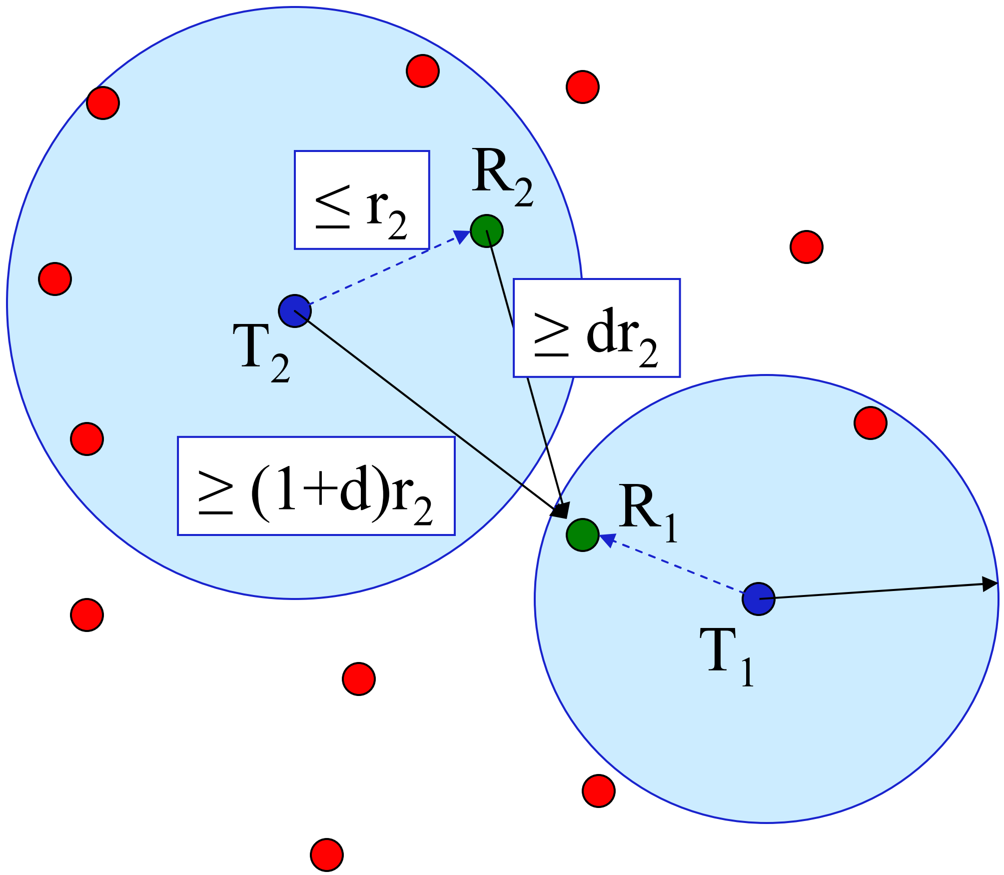
Maximum simultaneous transmissions



Shaded circles represent interference range

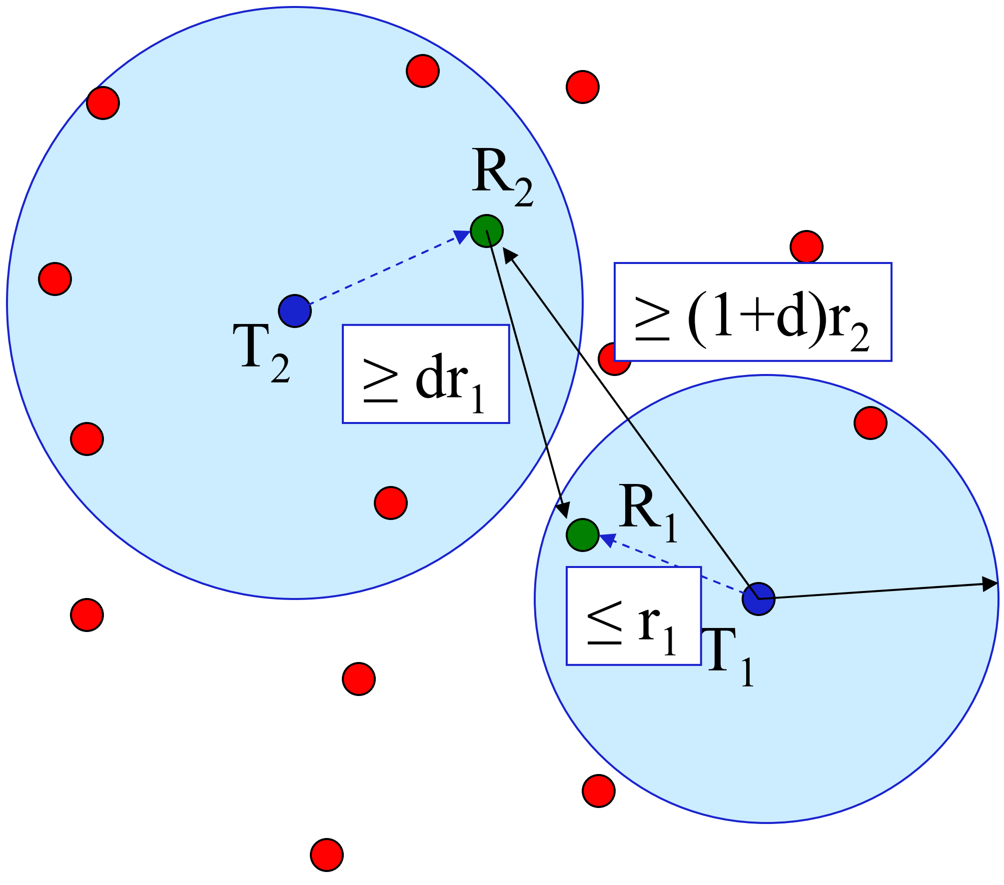
ACK: Slide adapted from PR Kumar’s plenary talk at Sensys 2004.

Maximum simultaneous transmissions



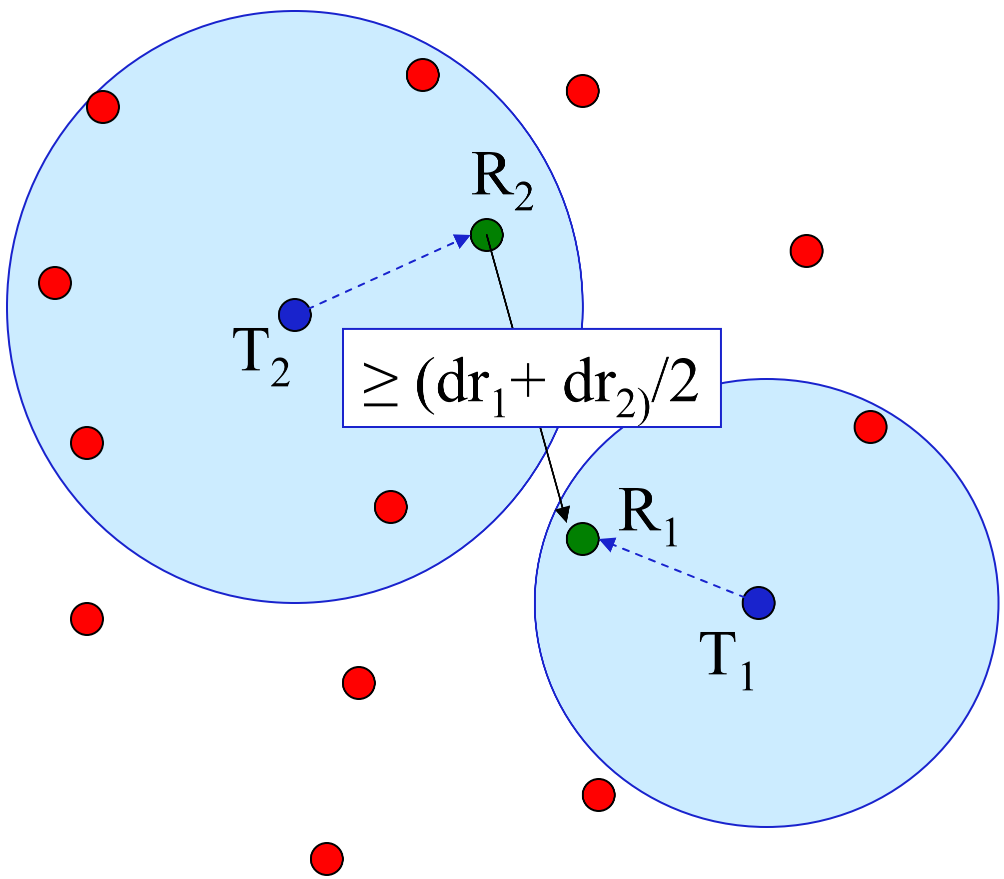
ACK: Slide adapted from PR Kumar's plenary talk at Sensys 2004.

Maximum simultaneous transmissions



ACK: Slide adapted from PR Kumar's plenary talk at Sensys 2004.

Maximum simultaneous transmissions



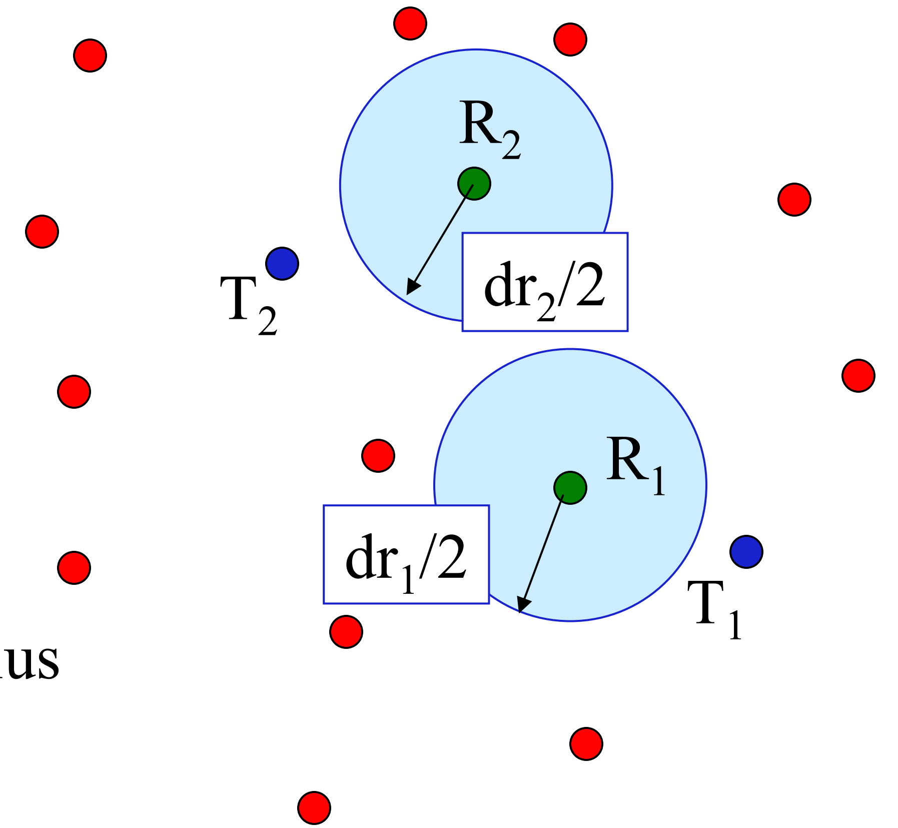
$R_1R_2 \geq dr_1$

$R_1R_2 \geq dr_2$

Add the two inequalities

ACK: Slide adapted from PR Kumar's plenary talk at Sensys 2004.

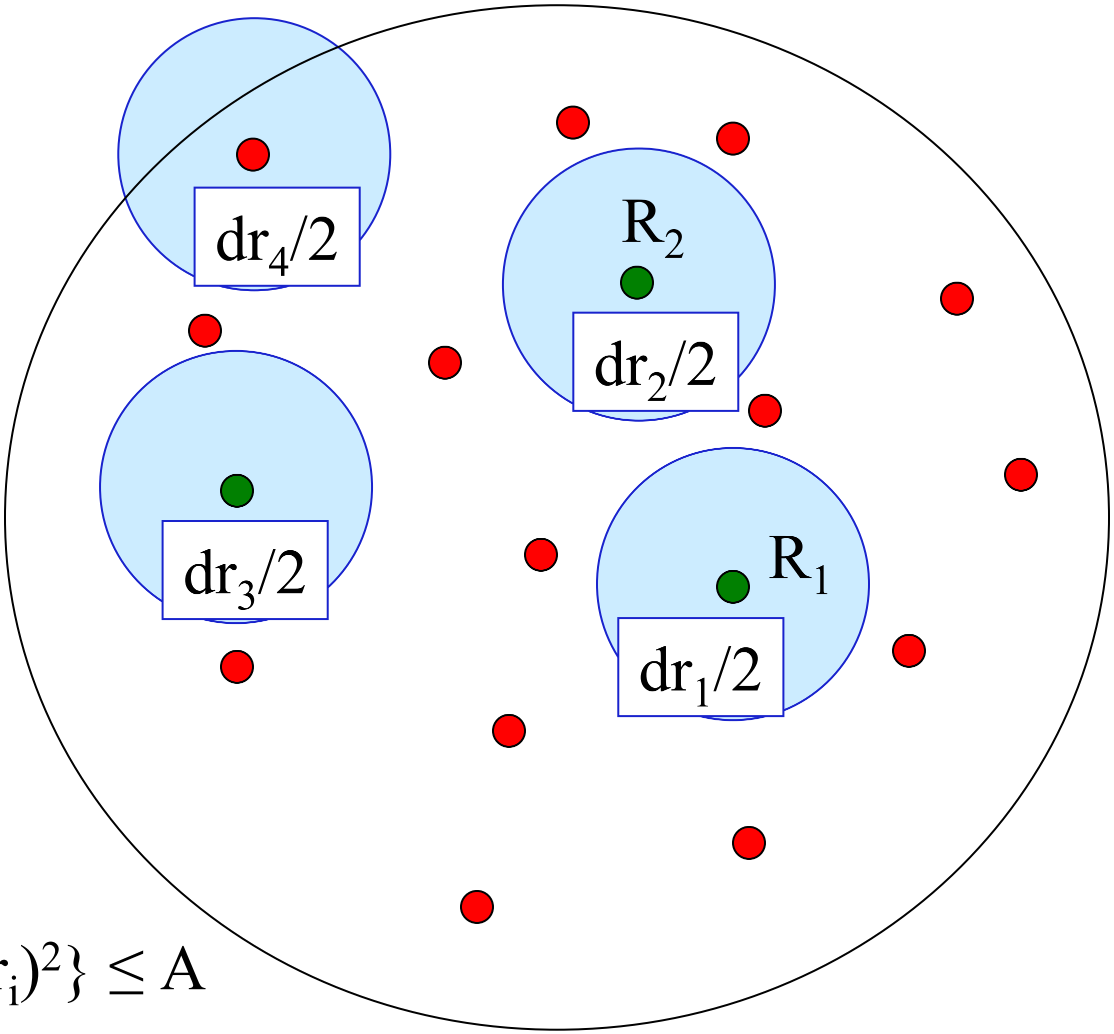
Maximum simultaneous transmissions



At receiver R_1, radius $dr_1/2$ is disjoint from any other reception

ACK: Slide adapted from PR Kumar's plenary talk at Sensys 2004.

Maximum simultaneous transmissions



$$\sum_{i=1}^{N/2} (1/4)\{\pi(dr_i)^2\} \leq A$$

ACK: Slide adapted from PR Kumar's plenary talk at Sensys 2004.

Maximum simultaneous transmissions

$$\sum_{i=1}^{N/2} (1/4)\{\pi(dr_i)^2\} \leq A$$

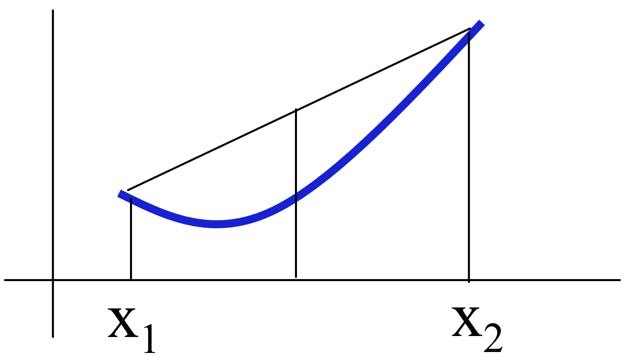
Rewrite:

$$\sum_{i=1}^{N/2} r_i^2 \leq 16A / \pi\, d^2$$

Convexity:

$$\frac{1}{N/2}\left[\sum_{i=1}^{N/2} r_i\right]^2 \leq \frac{1}{N/2}\sum_{i=1}^{N/2} r_i^2 \leq 32A / \pi\, Nd^2$$

Convexity:



$$f(x_1)/2+f(x_2)/2 \geq f(x_1/2+x_2/2)$$

r^2 is a convex function

ACK: Slide adapted from PR Kumar's plenary talk at Sensys 2004.

Transport Capacity

- Transport capacity per transmission is $W*r_i$ (bit meters/second)
- Total network transport capacity:

$$\sum_{i=1}^{N/2} Wr_i \leq W\,[8A / \pi N\, d^2]^{1/2}$$

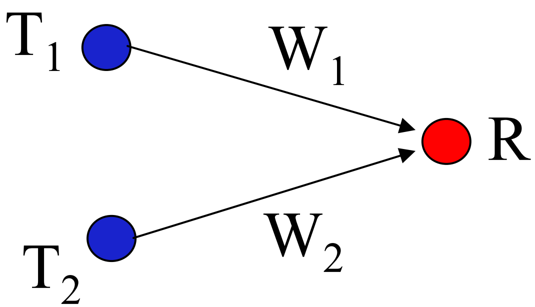
- Per node transport capacity: $O(\sqrt{1/N})$
- It can be proved that this is achievable as well
  - Requires multi-hop operation

ACK: Slide adapted from PR Kumar's plenary talk at Sensys 2004.

Design Implications

- Large network with arbitrary communication pattern is NOT scalable (per node capacity is diminishing)
  - Multi-hop operation can achieve the $O(\sqrt{1/N})$
- How to build a large network?

1. Use multi-user decoding: interference is also information!
   - Active area of research



2. Transmit only a relevant function of the data: in-network processing
3. Exploit correlation in data at multiple transmitters
4. Large systems use hierarchy: add infrastructure

In Network Processing

[Giridhar and Kumar 2005]

- Random multi-hop network, one fusion node
  - certain class of functions (mean, mode, std. deviation, max, k-th largest etc.) can be extracted at rate O{1/log(n)}
  - Exponentially more capacity than without in network processing: O(1/n)

- Network protocol design
  - Tessellate
  - Fuse locally
  - Compute along rooted tree of cells

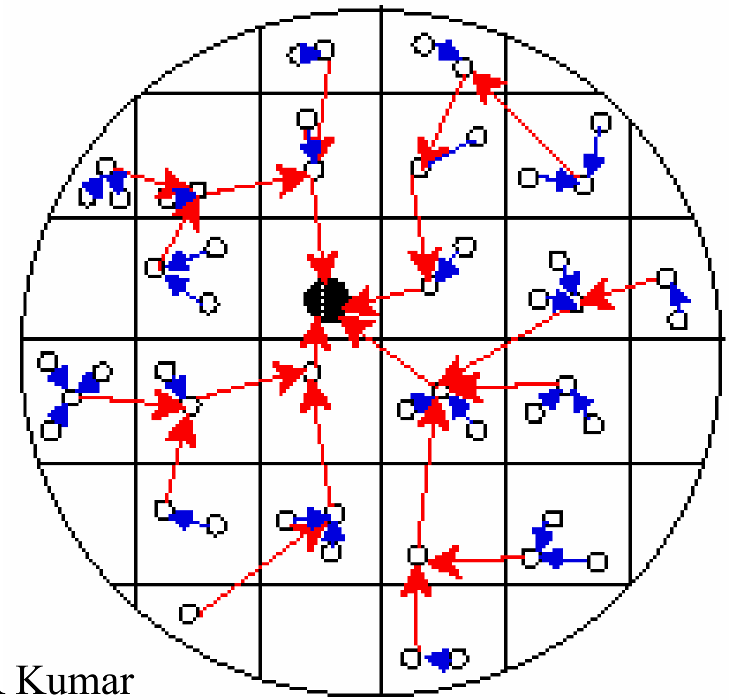


Fig. Ack: PR Kumar

Routing: Exploit Data Correlation

[Luo and Pottie 2005]

- Each sensor generates measurement of data size R
- Due to correlation, extra information at additional sensor has data size r

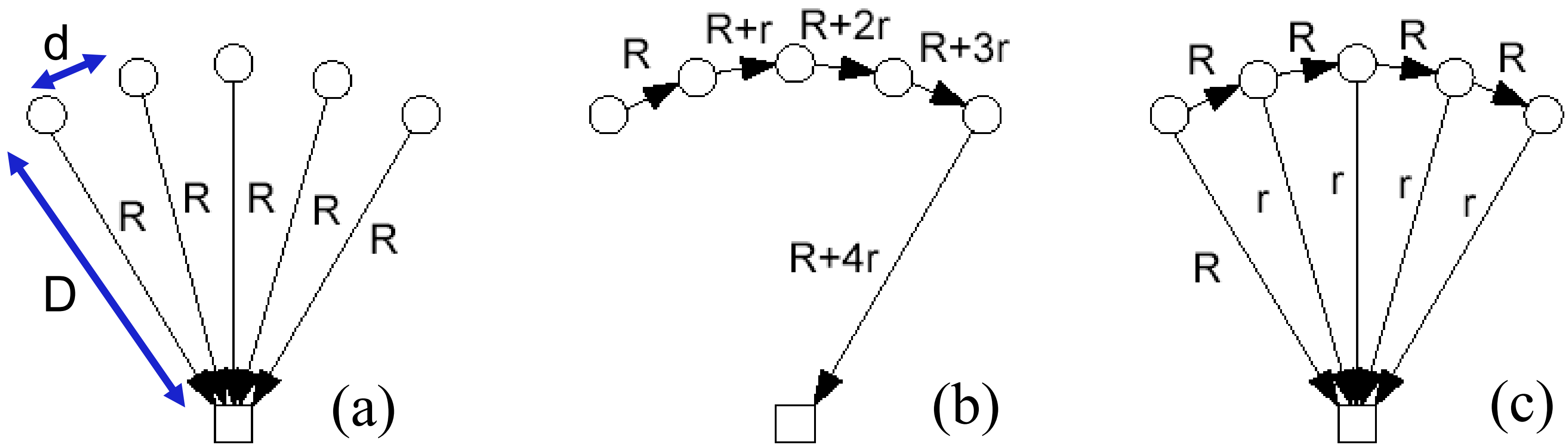


Fig. Ack: Huiyu Luo

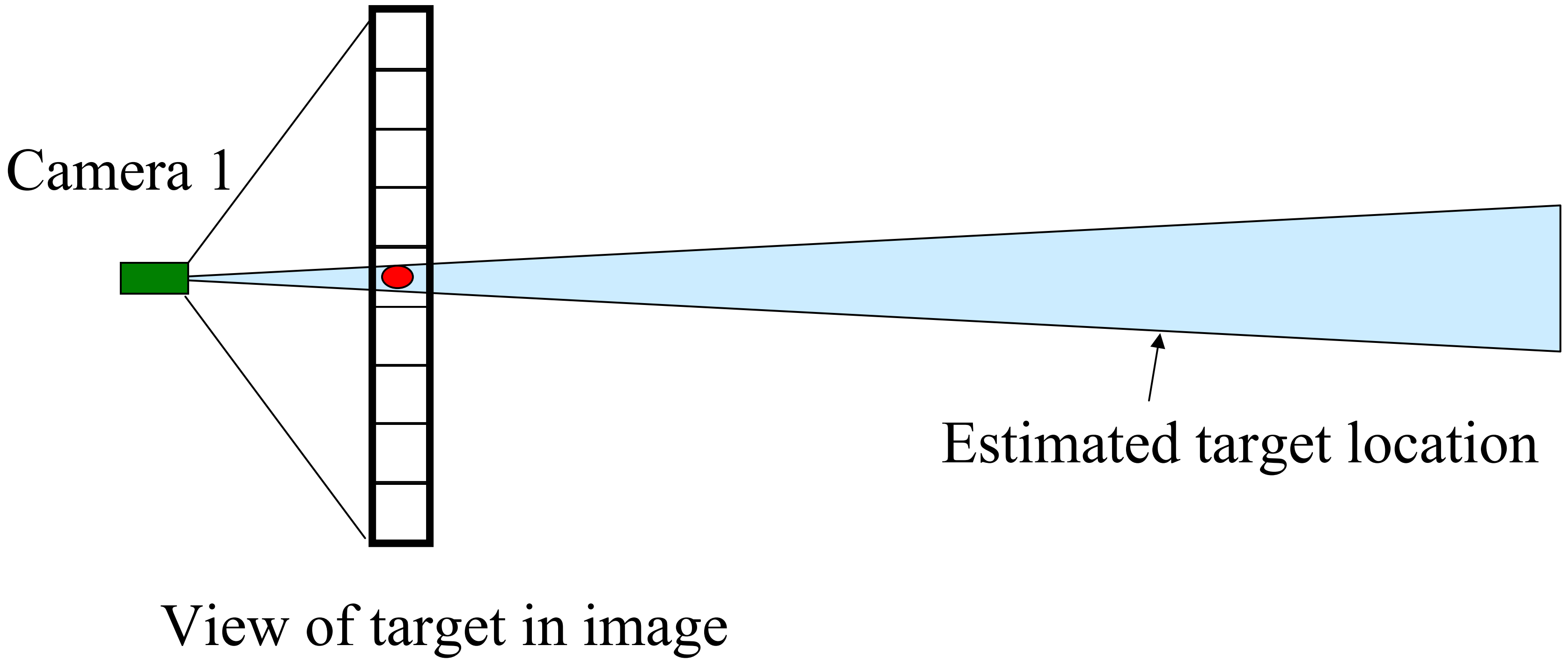
$$C_a = 5RD$$

$$C_b = RD + 4rD + 4Rd + 6rd$$

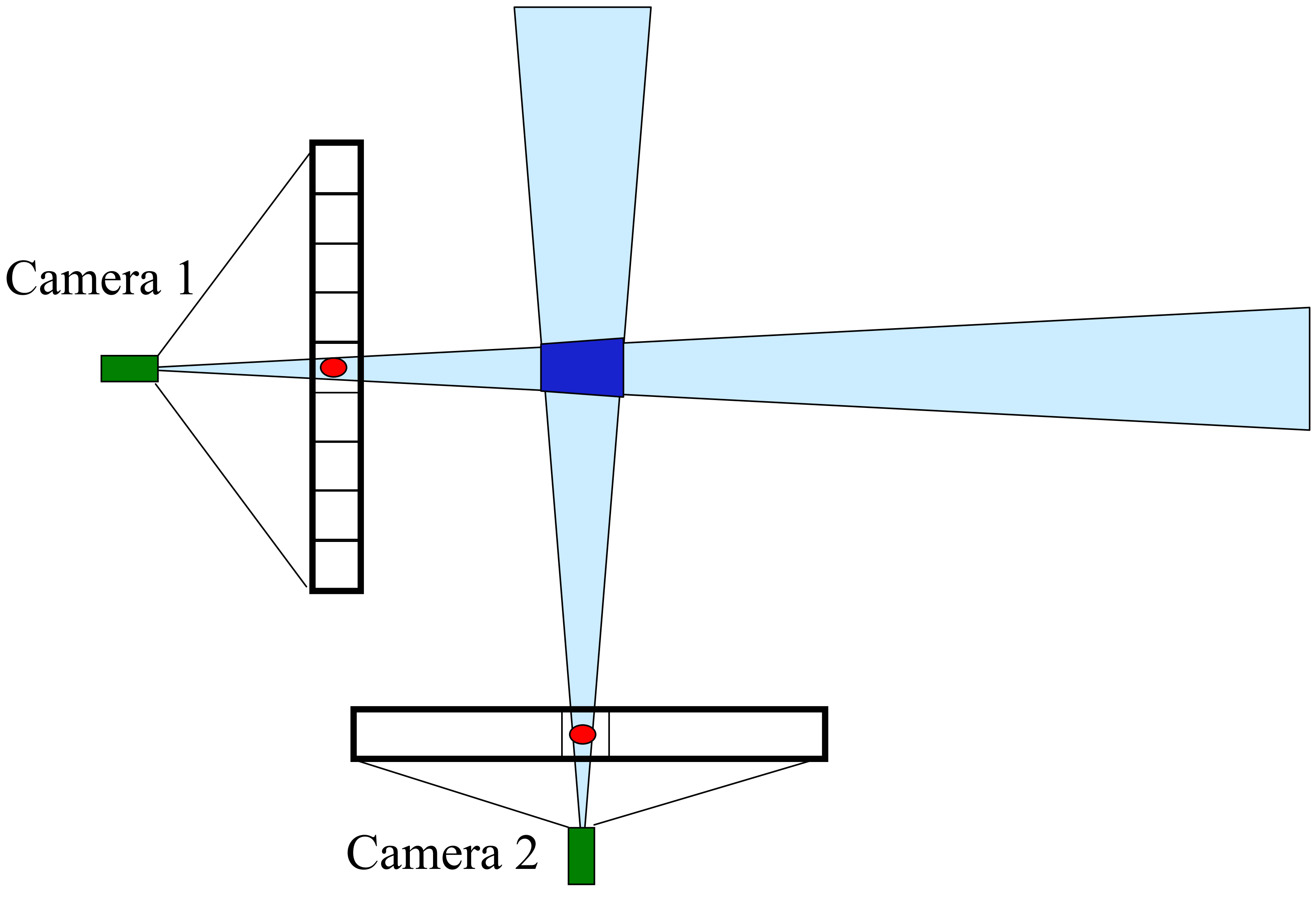
$$C_c = RD + 4rD + 4Rd$$

Data Fusion

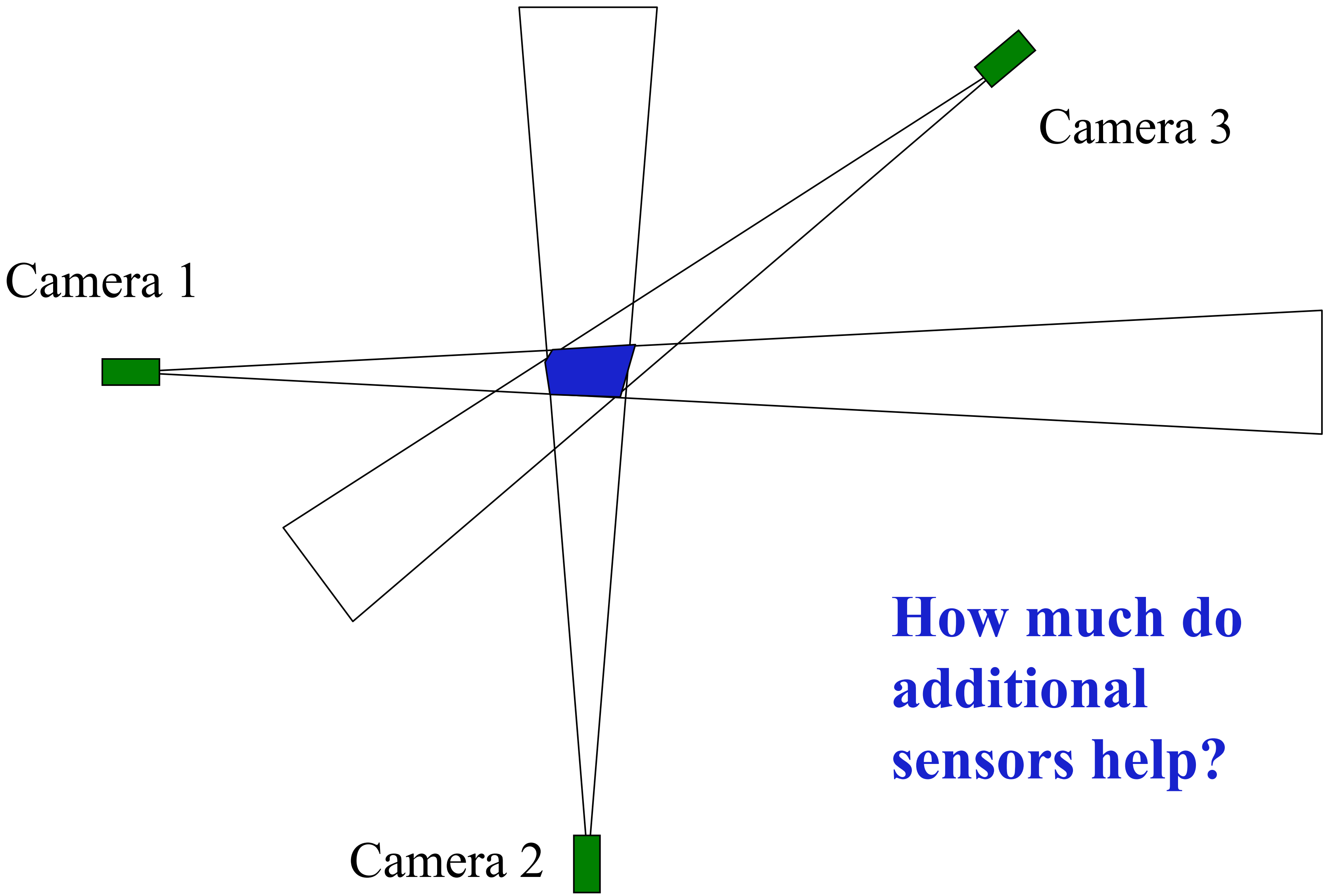
Fusing multiple sensors helps



Fusing multiple sensors helps



Fusing multiple sensors helps



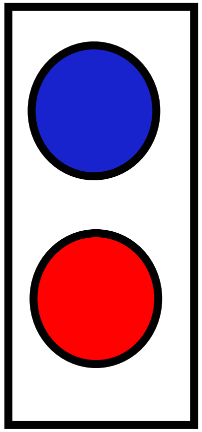
How much do additional sensors help?

Resort to Information Theory

- Model the information content of each sensor
- Measure the combined information content of multiple sensors assuming best possible fusion methods used
  - Eg. Compute the distortion achieved for a given data size
- Can then determine if the improvement in distortion is worth the extra sensors

Measuring Information

- Information depends on randomness
- Two balls: $P_{red} = 0.5$, $P_{blue} = 0.5$



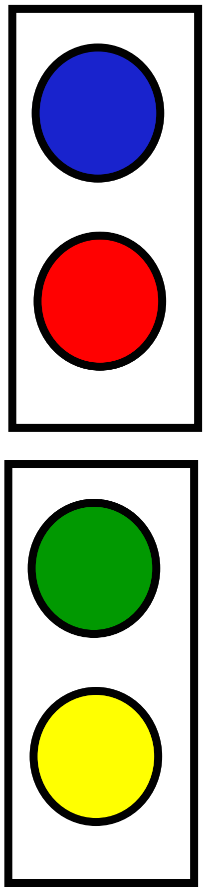
Let message be:
Red ball is chosen = 0,
Blue ball is chosen = 1

Message has 1 bit of information

Let
Red ball is chosen = RED,
Blue ball is chosen = BLU
Still only 1 bit of information

Measuring Information

- Four balls:
- $P_{red} = 0.25$, $P_{blue} = 0.25$, $P_{green} = 0.25$, $P_{yellow} = 0.25$

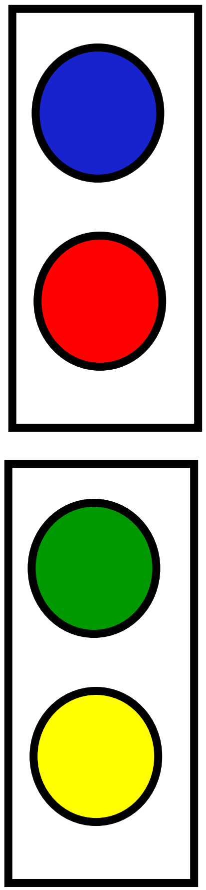


Let
Red = 00, Blue = 01, Green = 10, Yellow = 11

Which ball is chosen = 2 bits of information

Measuring Information

- Four balls, but unequal probabilities:
- $P_{red} = 0.5$, $P_{blue} = 0.25$, $P_{green} = 0.125$, $P_{yellow} = 0.125$



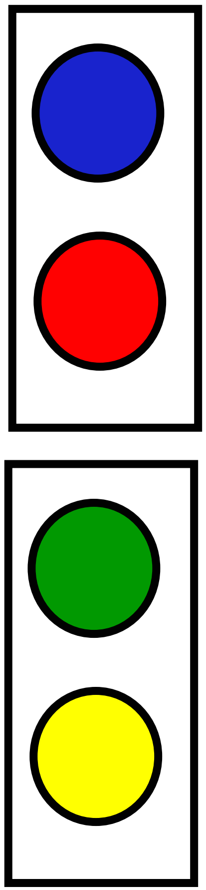
Save bits on more likely cases:
Red = 0, Blue = 10, Green = 111, Yellow = 110

Average number of bits to communicate:
$0.5*1 + 0.25*2 + 0.125*3 + 0.125*3 = 1.75$
(over many-many balls)

Which ball is chosen = 1.75 bits of information

Measuring Information

- Number of bits measured by ENTROPY
- $H = -\Sigma p_i * \log_2(p_i)$



Average number of bits to communicate:
0.5*1 + 0.25*2 + 0.125*3 + 0.125*3 = 1.75
(over many-many balls)

Entropy:
-[0.5*log(0.5) + 0.25*log(0.25) + 0.125*log(0.125)
+ 0.125*log(0.125)] = 1.75

Example: English alphabet

- Ignoring case, we have 27 characters
  - Naively: need 5 bits to represents a character
    - $2^4 = 16 < 27 < 2^5 = 32$
- Alternatively: measure the probabilities
  - calculate H ≈ 1.5 bits
- Efficient methods to automatically learn the probabilities and compress a file are available
  - Eg: Zip programs

i	a_i	p_i
1	a	0.0575
2	b	0.0128
3	c	0.0263
4	d	0.0285
5	e	0.0913
6	f	0.0173
7	g	0.0133
8	h	0.0313
9	i	0.0599
10	j	0.0006
11	k	0.0084
12	l	0.0335
13	m	0.0235
14	n	0.0596
15	o	0.0689
16	p	0.0192
17	q	0.0008
18	r	0.0508
19	s	0.0567
20	t	0.0706
21	u	0.0334
22	v	0.0069
23	w	0.0119
24	x	0.0073
25	y	0.0164
26	z	0.0007
27	–	0.1928

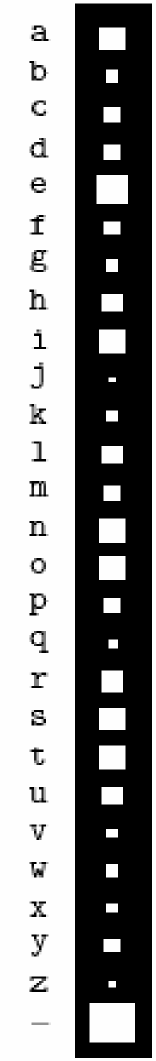
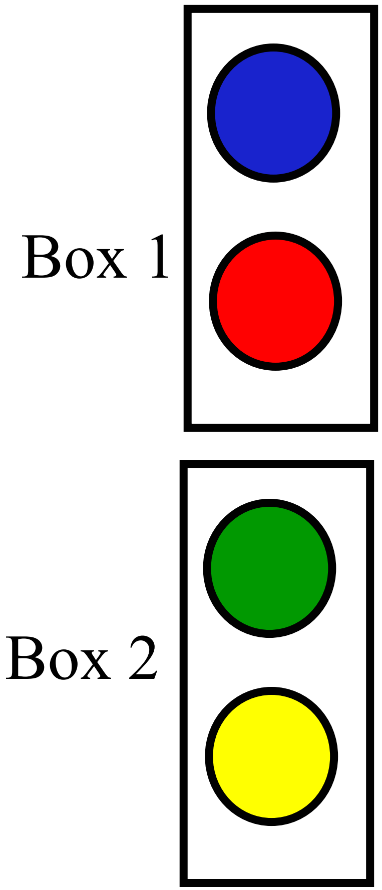


Fig Ack: Mackay 2003

Conditional Entropy

- What if only incomplete information is available?



Let $P_{red} = 0.25$, $P_{blue} = 0.25$, $P_{green} = 0.25$, $P_{yellow} = 0.25$

Given Box 1 is chosen:

$P(red|B1) = 0.5$, $P(blue|B1) = 0.5$,

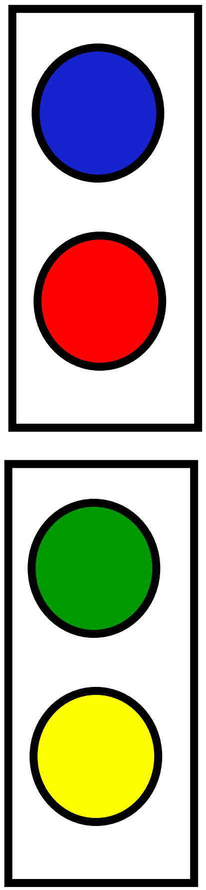
$P(green|B1) = 0$, $P(yellow|B1) = 0$

Now, which ball is chosen = 1 bit of information

Conditional Entropy

- Suppose variable y has partial information about x

Box 1



Box 2

$$H(x|y=a) = -\Sigma\, p(x_i|y=a)*\log_2(p(x_i|y=a))$$

Given Box 1 is chosen:

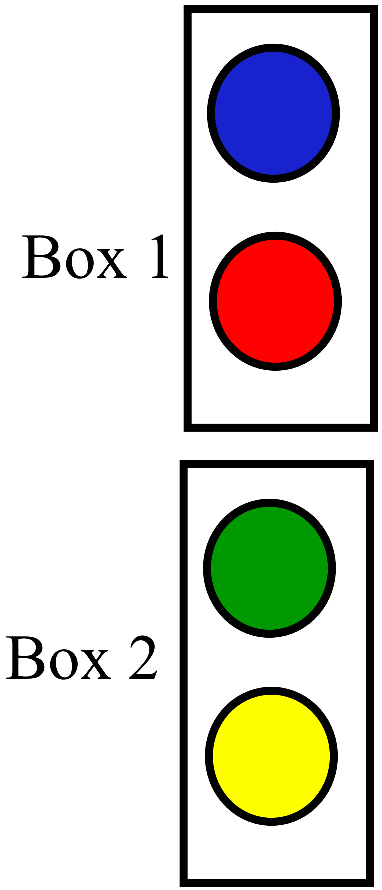
P(red|B1) = 0.5, P(blue|B1) = 0.5,

P(green|B1) = 0, P(yellow|B1) = 0

$$H(color|box=b1) = 0.5*\log(0.5) + 0.5*\log(0.5)$$
$$+0*\log(0) + 0*\log(0) = 1$$

Conditional Entropy

- Entropy can also be defined for conditional probabilities
  - Take weighted average with probabilities of y



$$H(x|y) = -\Sigma_a\, p(y=a)\, H(x|y=a)$$

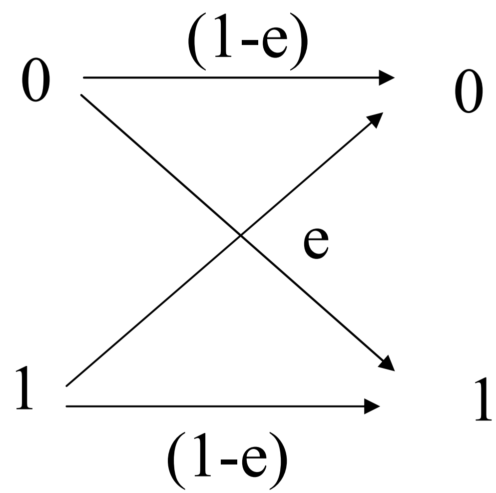
Given Box 1 is chosen:
P(box=B1) = 0.75, P(box=B2) = 0.25,

H(color|box) = 0.75*H(color|box=B1) + 0.25*
0.5*H(color|box=B2)
= 0.75*1 + 0.25*1 = 1

Conditional entropy = 1 bit of information

Example: Unreliable hard disks

- Conditional entropy helps model noise or channel error
- Given a hard disk with error probability = e



- Given e = 0.1
- Suppose: want more reliable HDD

Ack: Example adapted from Mackay 2003

Example: Unreliable hard disks

- Two Options:
  - Technology approach: Invent better storage, use more expensive components
  - Systems approach: build a reliable system around the unreliable HDD

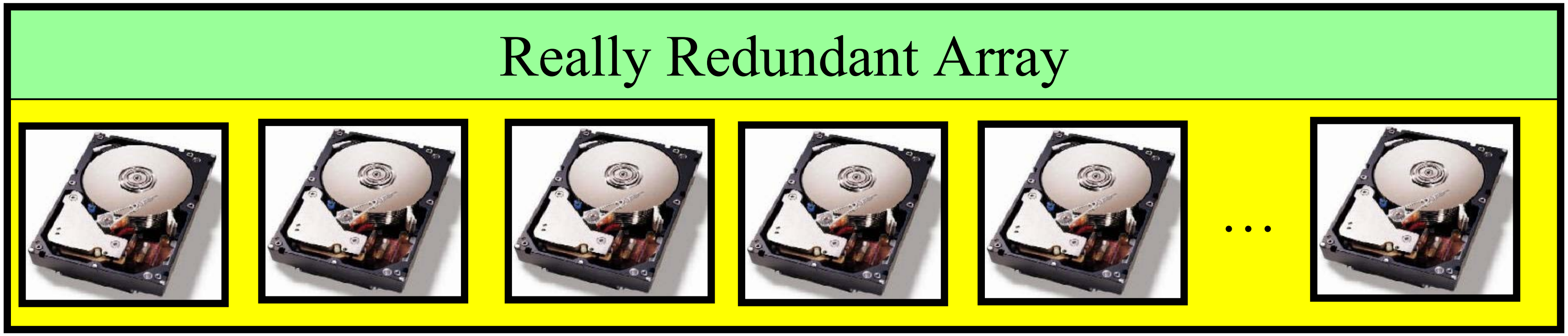
- Simple Strategy: Store every bit 3 times - 000, 111
  - if one bit corrupted, can still fix it
  - Error probability:
    $\underbrace{3*e^2(1-e)}_{\text{2 bits in error, 3 possible ways}} + \underbrace{e^3}_{\text{all 3 bits in error}} = 0.028$

2 bits in error,
3 possible ways

all 3 bits in error

Example: Unreliable hard disks

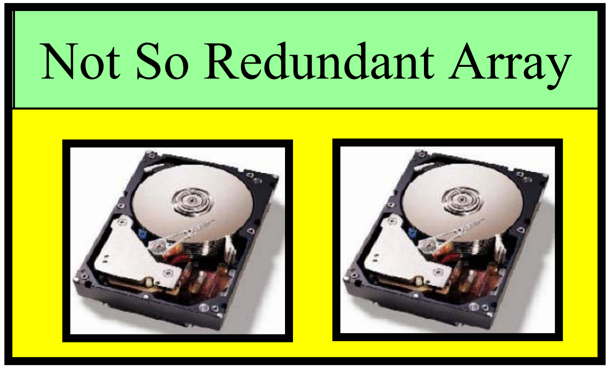
- Not happy with 2.8% error
  - Want error probability: 10^{-15}
  - Using simple strategy, need 60 disks
    - Not good for embedded design!



- Alternative: use conditional entropy analysis

Example: Unreliable hard disks

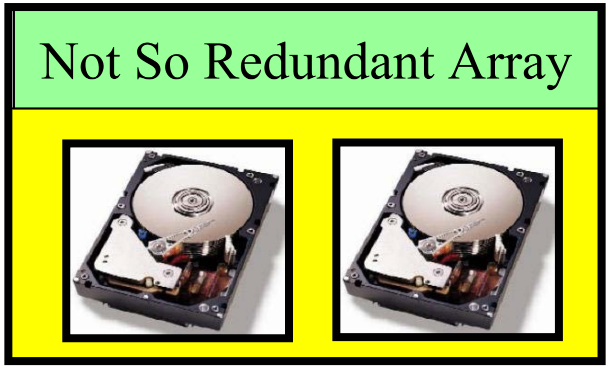
- Recovered data has errors
  - Has partial information about original data
- Denote
  - Y = recovered data
  - X = original data



- HDD communicates:
  - Information in X – information still left in X given Y
  - H(X) – H(X|Y)
    - Suppose source was $P_{red}=P_{blue}=0.5$, so H(X)= 1 bit
    - From error model know: P(X|Y). Gives H(X|Y) = 0.47
  - HDD returns 0.53 bits for each bit stored
    - Two HDD's suffice! (can store 1.06 bits > 1 bit)

Example: Unreliable hard disks

- Recovered data has errors
  - Has partial information about original data
- Denote
  - Y = recovered data
  - X = original data



- HDD communicates:
  - Information in X – information still left in X given Y
  - $H(X) - H(X|Y)$

CAVEAT: This holds only when a very large amount of data is stored.
Simple repetition strategy does not work- better coding needed.

Mutual Information

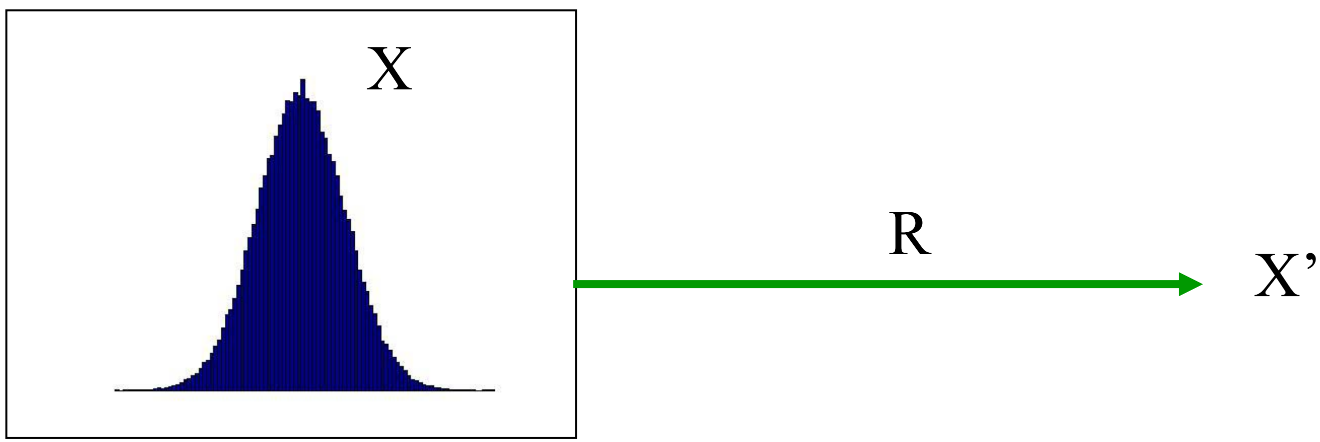
- The quantity H(X) – H(X|Y) is known as MUTUAL INFORMATION
- Written as I(X;Y) = H(X) – H(X|Y)
  - Helpful for measuring “recovered information” in many problems
  - Eg. Capacity of a channel (such as a hard disk or a wireless channel)
    - $\max_{P(X)} I(X;Y)$

Rate Distortion

- Suppose the HDD had zero error probability but we stored more data than its capacity
  - Can think of it as storing with loss due to error: again characterized using mutual information
  - Using lesser bits than information content in original data
    - Eg. JPEG, MP3
- Given a distortion constraint D, how much data, R, is required to be stored:
  - $R(D) = \min I(X;X')$
    - X' is such that it satisfies the distortion constraint and minimizes the mutual information

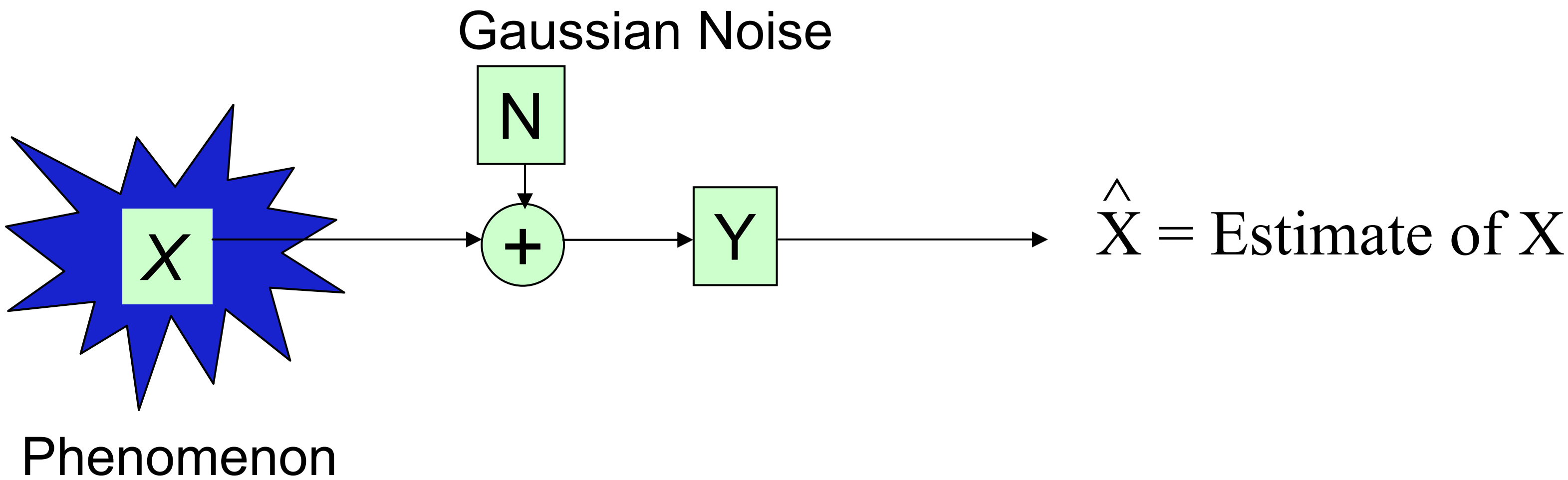
Gaussian Phenomenon

- Rate distortion function has been calculated when X is a Gaussian random variable
  - X = Gaussian with mean = 0, std dev = σ
  - R(D) = (1/2)*log(σ^2/D)
    - for D ≤ σ^2. If tolerable distortion more than phenomenon variance, no need to send any data



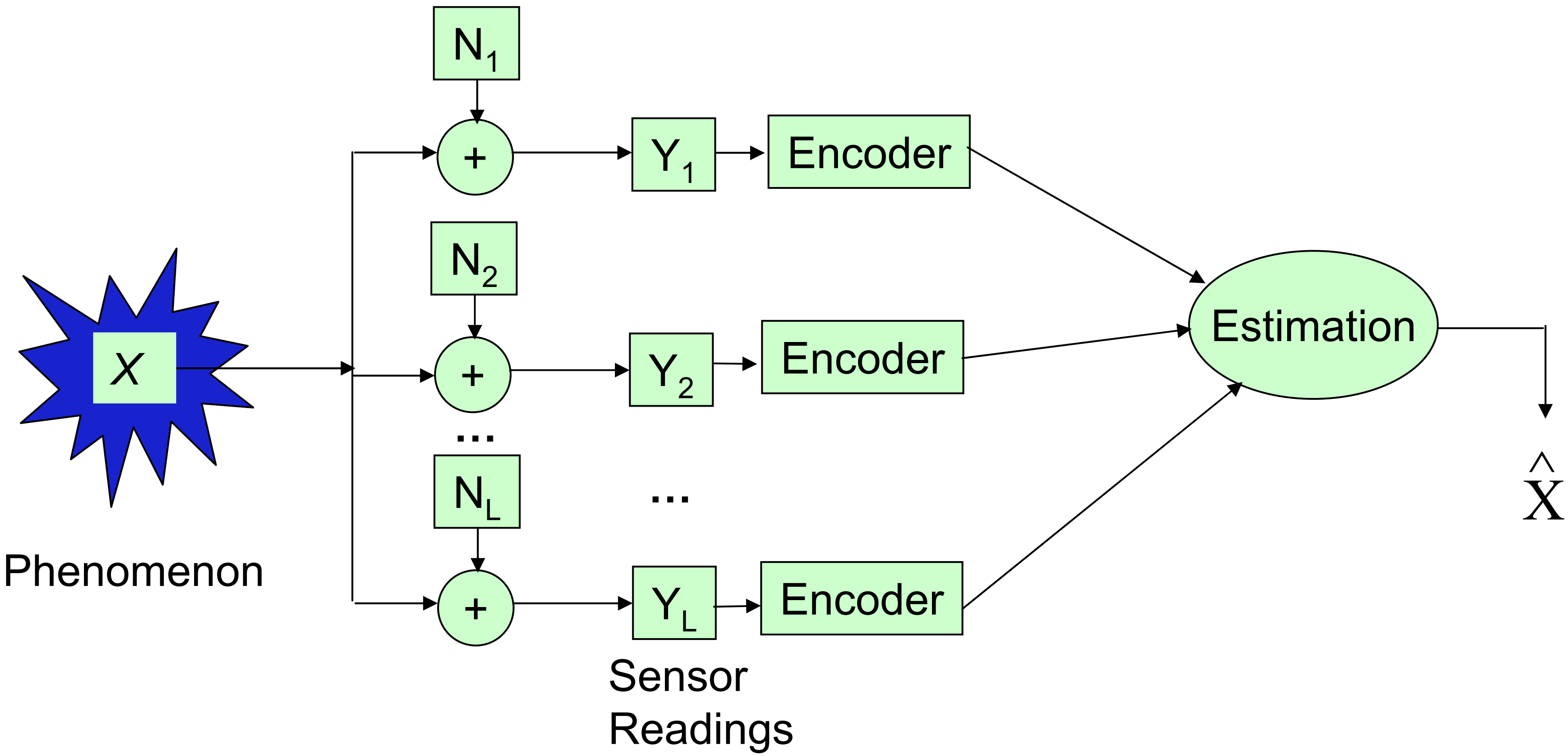
Proof: [Thomas and Cover]

Effect of Sensing Noise



- For given data rate, what's the distortion achievable?
  - R(D) measure with noise:
  - $R(D) = (1/2)*\log [\sigma^4 / (\sigma^2 D + \sigma_N^2 D - \sigma^2 \sigma_N^2)]$
    - When positive, else zero rate

Quantifying the Fusion Advantage



- Characterize the distortion achieved with given rate when multiple sensors used
  - Distortion advantage gives the benefit of additional sensors

Quantifying the Fusion Advantage

[Chen et al. 2004]

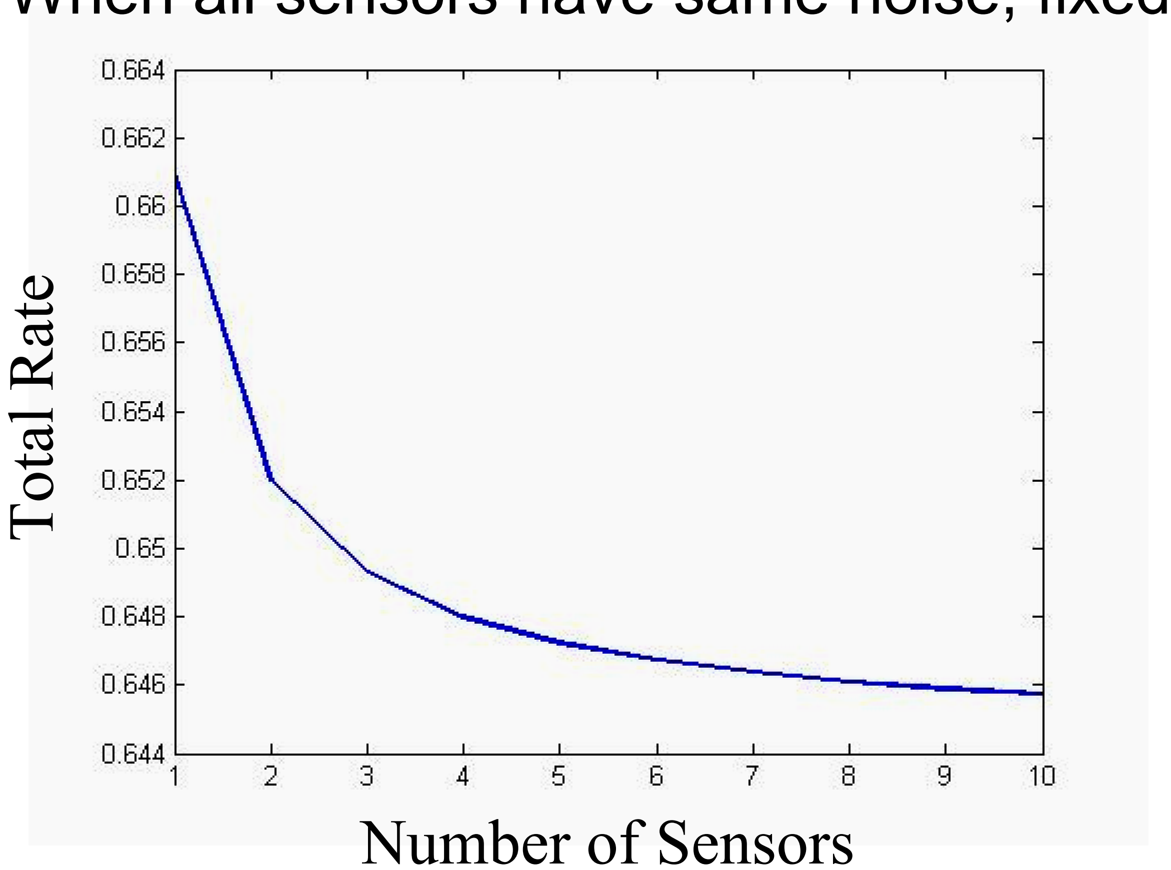
- The total rate generated at multiple sensors is related to the distortion in estimation as:

$$\sum_{i=1}^{L} R_i = \frac{1}{2}\log^{+}\left\{\frac{\sigma_X^2}{D}\left(\frac{D\sigma_X^2 L}{D\sigma_X^2 L - \sigma_X^2\sigma_N^2 + D\sigma_N^2}\right)^{L}\right\}$$

  - where L is the number of sensors, σ_X^2 is the variance of X and σ_N^2 is the noise variance

Design Implications

- When all sensors have same noise, fixed D



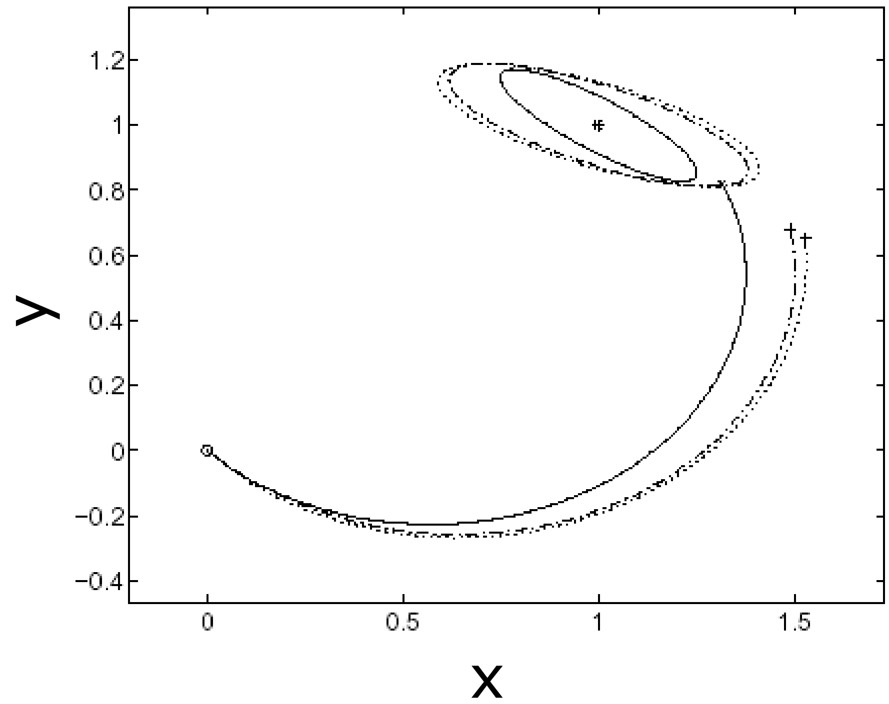
This is for Gaussian noise and Gaussian phenomenon

- Using multiple sensors helps
  - Returns diminish beyond a small number of sensors

Another application of Mutual Information

[Grocholsky 2002]

- Mobile sensor: Having taken a measurement, where to take the next measurement?
- Among all possible motion steps, choose one which maximizes mutual information between phenomenon and measurement
  - Assume that density of phenomenon and noise model for sensor known
- Also extended to teams of mobile sensors

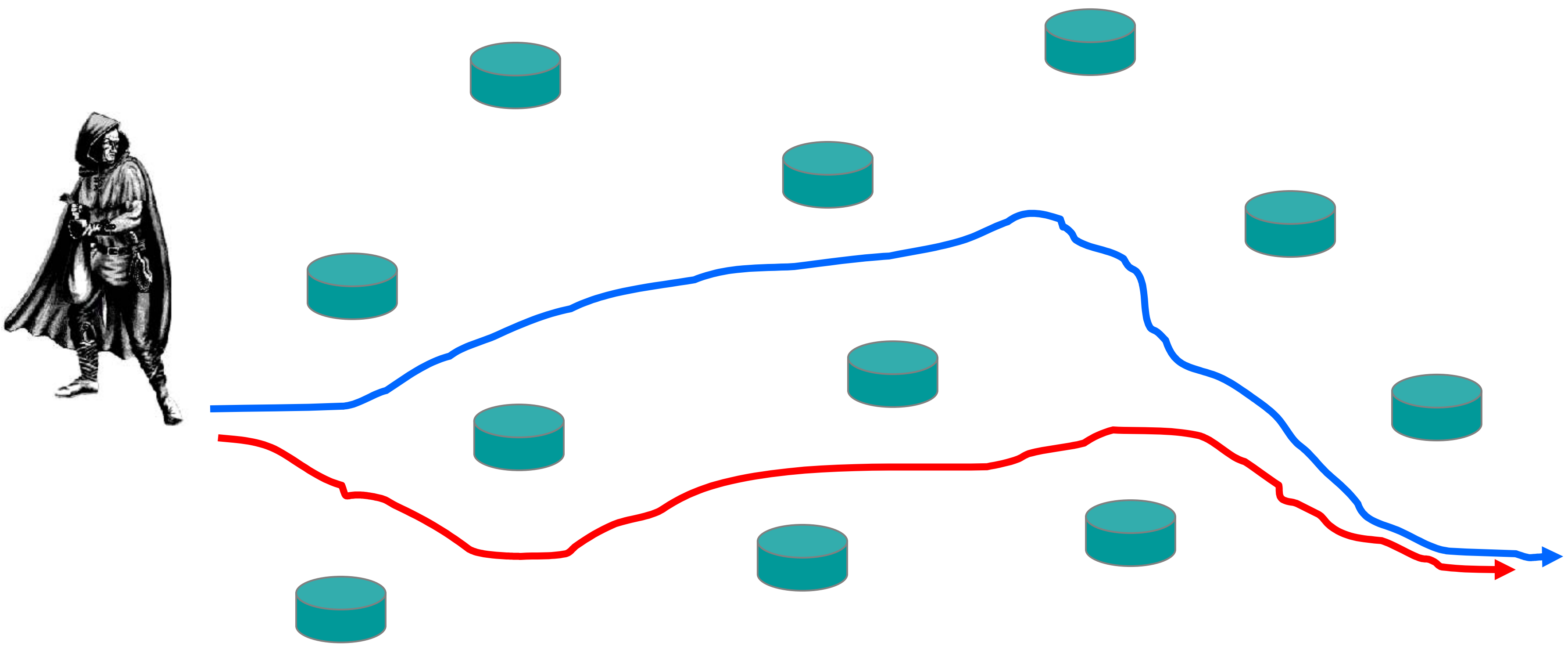


Coverage and Deployment

Worst-case Coverage

[Meguerdichian et al, 2001]

- Minimal exposure path
  - Likelihood of detection depends on distance from sensor and time in range



Sensing Model

Sensing model *S* at an arbitrary point *p* for a sensor *s* :

$$S(s,p) = \frac{\lambda}{\left[d(s,p)\right]^K}$$

where *d(s,p)* : distance between the *s* and *p*
constants λ and *K* are technology and environment dependent

Slide adapted from Meguerdichian, Mobicom 2001]

Intensity Model

Effective sensing intensity at point *p* in field *F* :

Aggregate of all sensors

$$I_A(F,p) = \sum_{1}^{n} S(s_i, p)$$

(Can generalize for fusion based metrics)

Closest Sensor

$$s_{\min} = s_m \in S \,\middle|\, d(s_m, p) \le d(s, p) \quad \forall s \in S$$
$$I_C(F,p) = S(s_{\min}, p)$$

(Can generalize for k closest sensors)

Slide adapted from Meguerdichian, Mobicom 2001]

Definition: Exposure

The ***Exposure*** for an object *O* in the sensor field during the interval [t_1, t_2] along the path *p(t)* is:

$$E\left(p(t),t_1,t_2\right)\stackrel{\Delta}{=}\int_{t_1}^{t_2} I\left(F,p(t)\right)\left|\frac{dp(t)}{dt}\right|dt$$

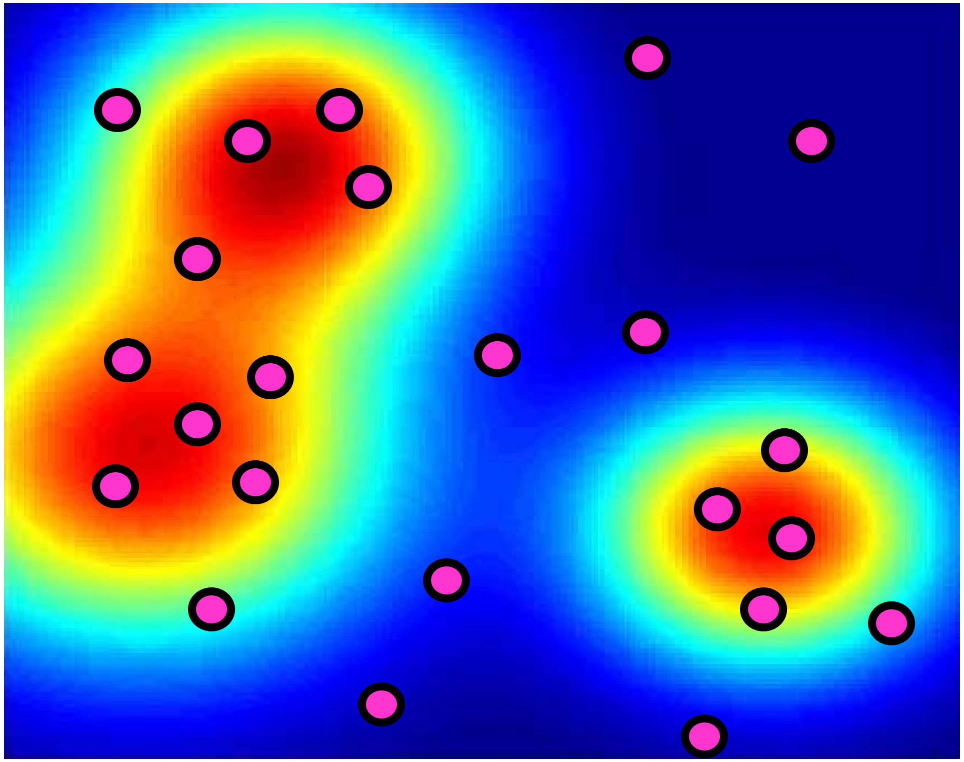
Methods available to compute this for a given network (using grid approximations and graph search techniques)

Slide adapted from Meguerdichian, Mobicom 2001]

Average Coverage

[Cortes et al, 2005]

- Quantify coverage with respect to phenomenon distribution in covered region
  - sensors close to phenomenon contribute more to coverage



= sensor

Shade represents phenomenon density

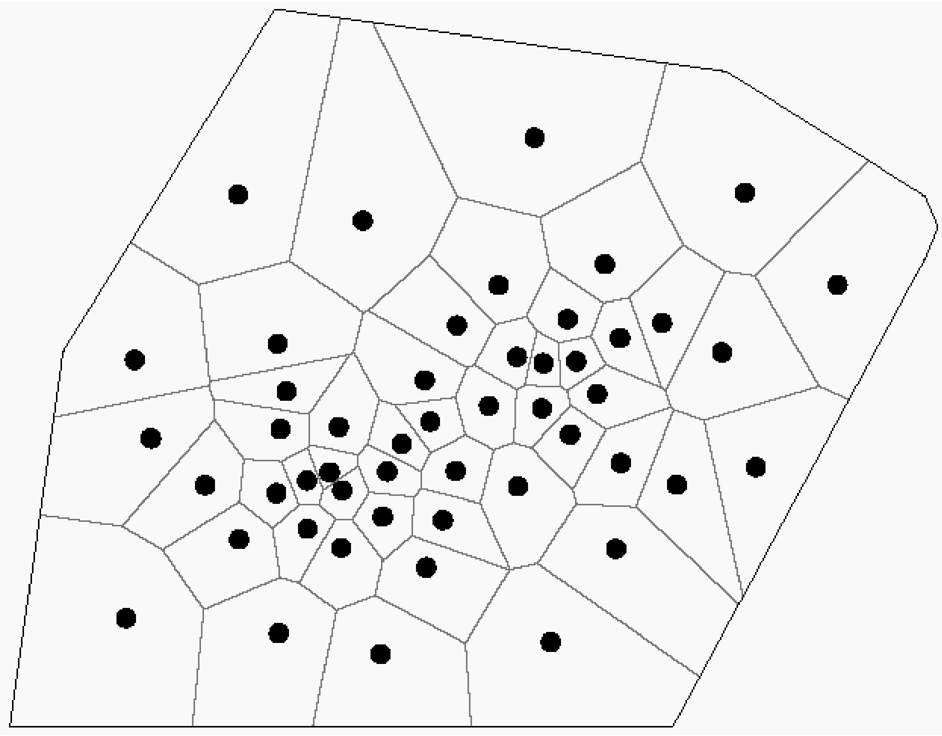
Average Coverage

- Region to cover: Q
  - q is a point in region Q
- Set of sensors: p_i, i=1,…,n
- Coverage function: f, depends on distance from sensor
  - Coverage at point q due to sensor p_i is $f(|q-p_i|)$
- Average coverage then becomes
  - Considering best sensor for each point q in Q

$$\mathcal{H}(P) = \underbrace{\int_Q}_{\text{Integrate over region covered}} \underbrace{\max_{i\in\{1,\dots,n\}} f(\|q-p_i\|)}_{\text{Coverage at point q due to best sensor}} \underbrace{\phi(q)}_{\text{Weight by phenomenon density}} dq$$

Example

- Suppose sensing quality declines with square of distance
  - This gives $f(x) = -x^2$
- Best sensor for a point q is thus the closest sensor to it
  - p_i responsible for points that are closer to p_i than any other sensor
  - Call such a region around p_i as the VORONOI cell of p_i



Voronoi cells

Example

- Average coverage for this *f* can be written as:
  - Integral over each Voronoi cell
  - Sum over all cells

$$\mathcal{H}_{\mathrm{C}}(P) = -\sum_{i=1}^{n} \int_{V_i(P)} \|q - p_i\|^2 \phi(q) dq$$

  - Where $V_i(P)$ is the Voronoi cell of p_i
  - Negative sign indicates that coverage reduces if distance from sensors is large

Planning Deployment

- Choose sensor locations to maximize coverage:
  - Eg. 1: Minimize worst case coverage by maximizing the exposure on minimum exposure path

$$E\left(p(t), t_1, t_2\right) \stackrel{\Delta}{=} \int_{t_1}^{t_2} I\left(F, p(t)\right) \left| \frac{dp(t)}{dt} \right| dt$$

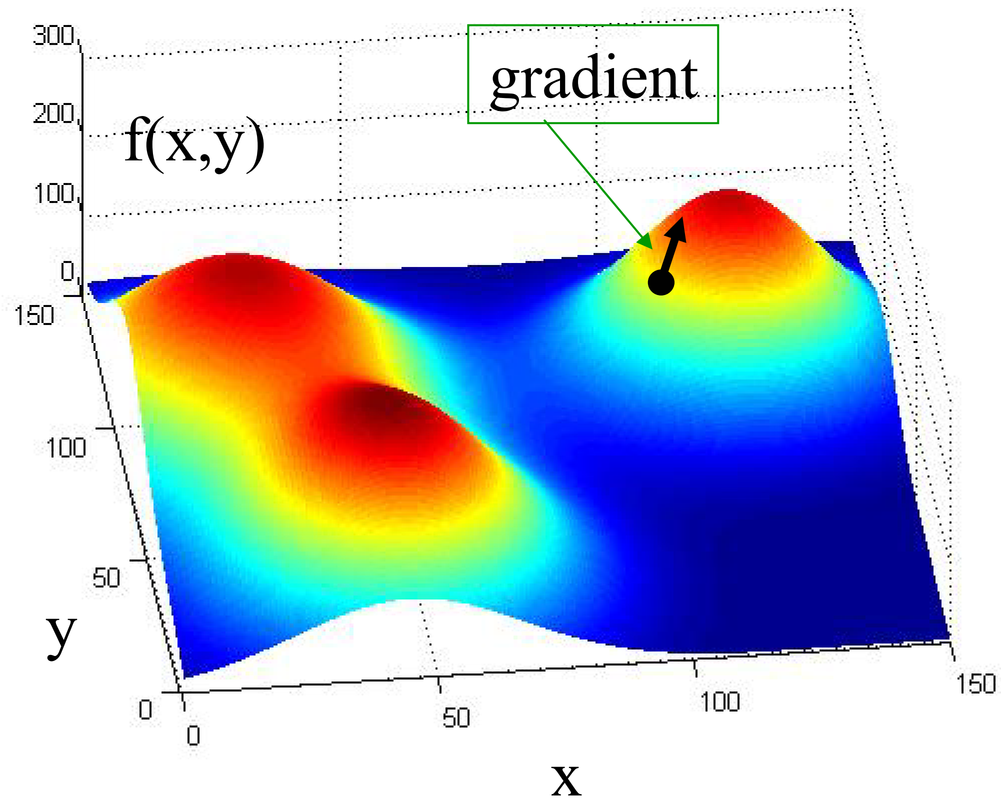
  - Eg. 2: Maximize $\mathcal{H}(P)$

Adapting Deployment at Real Time

- Suppose nodes are mobile
- Need distributed algorithm that allows nodes to change their locations in order to maximize coverage
  - Algorithm should use only local information and not the global view of the network
- One method to compute optima: gradient descent

Gradient Descent

- Consider a function f of two variables- x, y
- Find the x,y where f(x,y) is minimal
  - start at a random x,y
  - Compute gradient at that (x,y): gives direction of maximum change. (steepest slope)
  - Change (x,y) in that direction
- May reach local minima



Distributed Method

- Take the gradient of the average coverage

$$\mathcal{H}_{\mathrm{C}}(P) = -\sum_{i=1}^{n} \int_{V_i(P)} \|q - p_i\|^2 \phi(q) dq$$

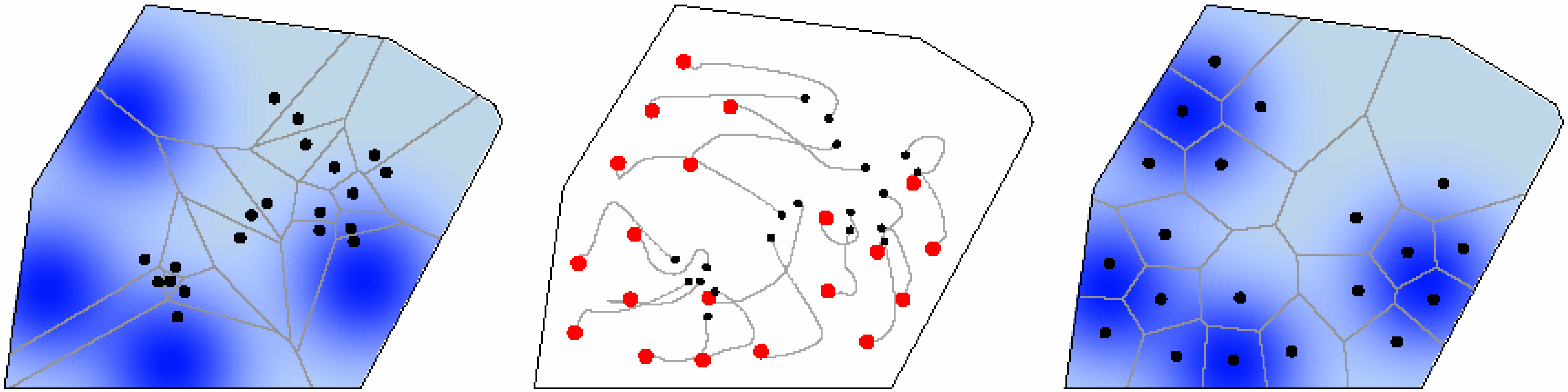
- Gradient turns out to be distributed over Voronoi cells!

$$\frac{\partial \mathcal{H}}{\partial p_i}(P) = 2 \int_{V_i(P)} (q - p_i) \phi(q) dq = 2\, \mathrm{M}_{V_i(P)} (\mathrm{CM}_{V_i(P)} - p_i)$$

  - where M is the mass of phenomenon density over the Voronoi cell of p_i and CM is the center of mass.

Distributed Method

- Distributed Motion algorithm: At each node p_i
  - Find Voronoi cell
  - Move towards center of mass
- Automatically optimizes total network coverage
  - No central coordination needed



Initial Gradient descent Final

Fig. Ack: F Bullo, April 15, 2005 talk at UCB

Energy Performance

Maximizing the lifetime

[Bhardwaj and Chandrakasan 2002]

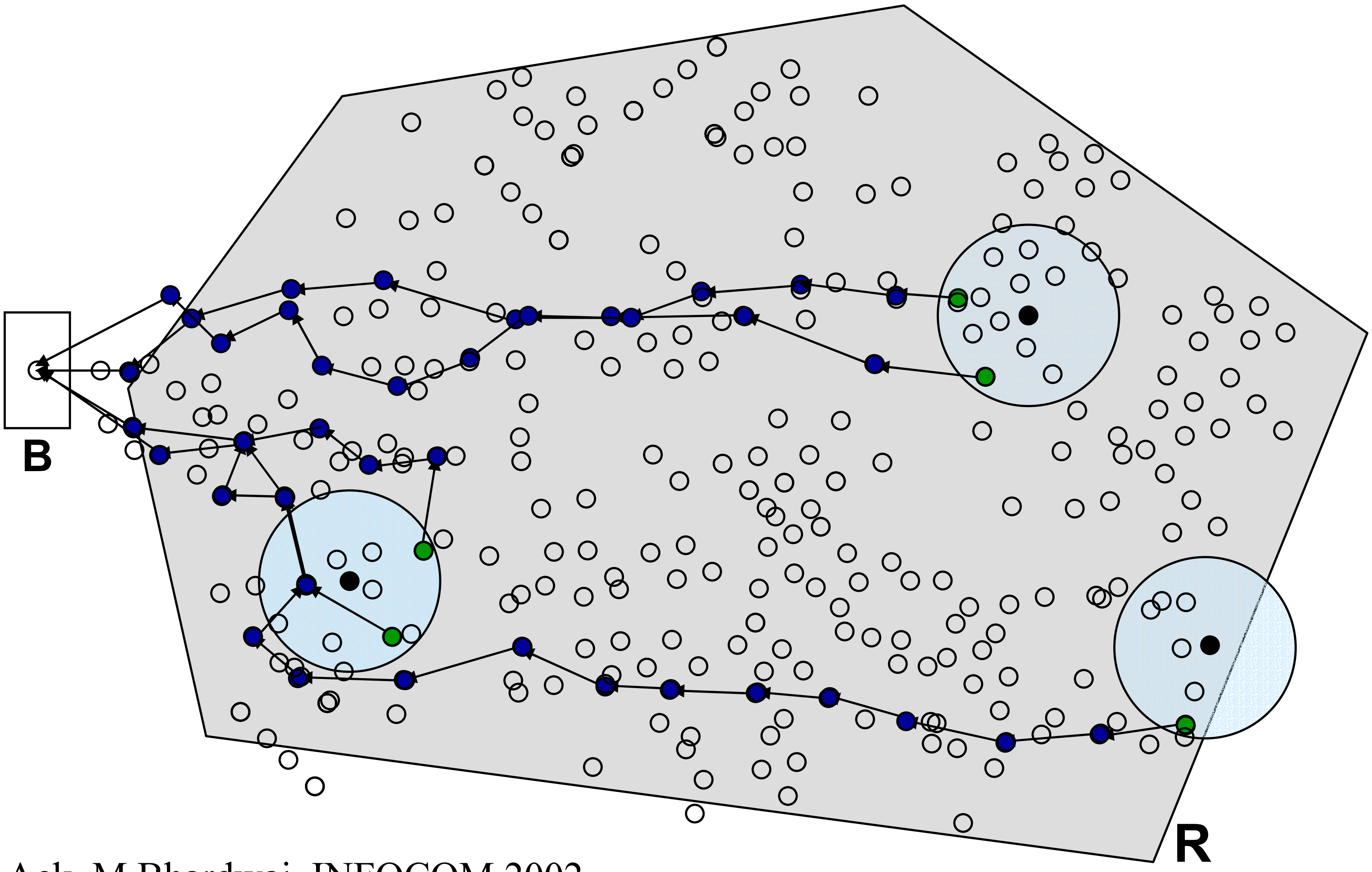
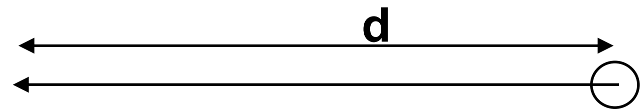


Fig Ack. M Bhardwaj, INFOCOM 2002

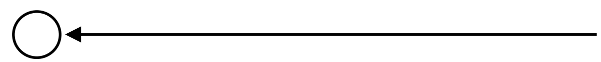
Problem Formulation

- Given
  - Set of data sources and intended destinations
  - Fixed battery levels at each node
- Find the best routing strategy: maximum lifetime for supporting required data transfers
  - Nodes can change their power level and choose best multi-hop paths

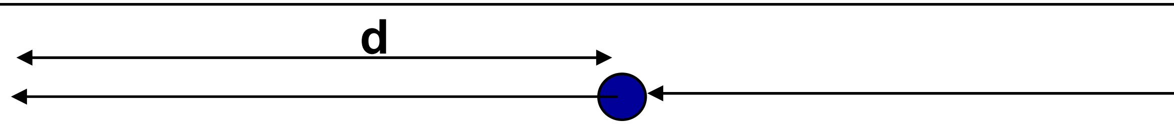
Energy Models



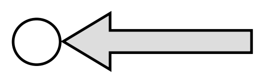
Transmit Energy Per Bit $E_{tx} = \alpha_{11} + \alpha_2 d^n$ n = Path loss index



Receive Energy Per Bit $E_{rx} = \alpha_{12}$



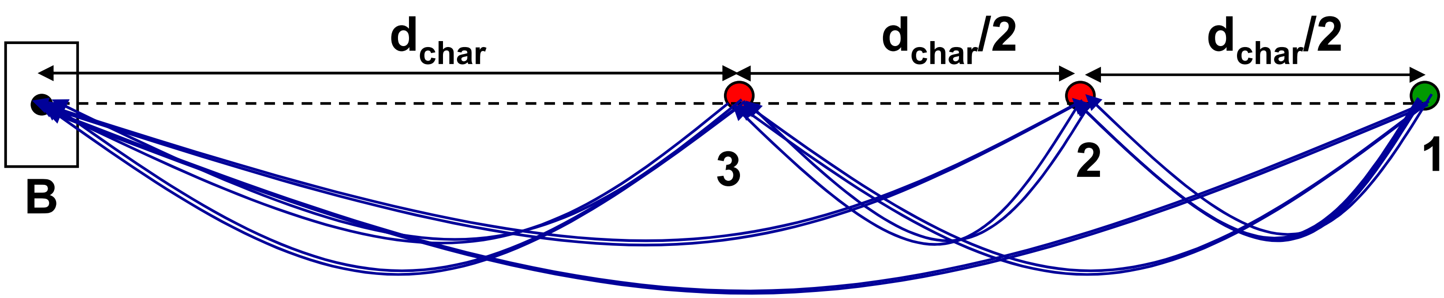
Relay Energy Per Bit $E_{relay} = \alpha_{11} + \alpha_2 d^n + \alpha_{12} = \alpha_1 + \alpha_2 d^n$



Sensing Energy Per Bit $E_{sense} = \alpha_3$

Slide adapted from: M Bhardwaj, INFOCOM 2002

Example: a three node network



- Ignore wasteful (loopy) routes, route choices are:

R_0: 1 → B
R_1: 1 → 2 → B
R_2: 1 → 3 → B
R_3: 1 → 2 → 3 → B

	Min-hop	Min-Energy		Optimal
	R_0: 0.25 R_1: 0 R_2: 0 R_3: 0	R_0: 0 R_1: 0 R_2: 1.0 R_3: 0	Other Choices…	R_0: 0 R_1: 0.375 R_2: 0.375 R_3: 0.625
Lifetime	**0.25**	**1.0**		**1.38**

Slide adapted from: M Bhardwaj, INFOCOM 2002

How to find the optimal choice?

- The search over multiple route choices can be stated as a linear program (LP)
  - Efficient tools exist for LP's

Objective :

$$\max \qquad t = \sum_{i=1}^{|F|} t_i$$

t_i = time for which route r_i is used, $|F|$ = number of choices

Constraints :

$$t_j \geq 0 \quad : \qquad 1 \leq j \leq |F|$$

$$\sum_{j=1}^{|F|} p(i, r_j) t_j \leq e_i \quad : \qquad 1 \leq i \leq N$$

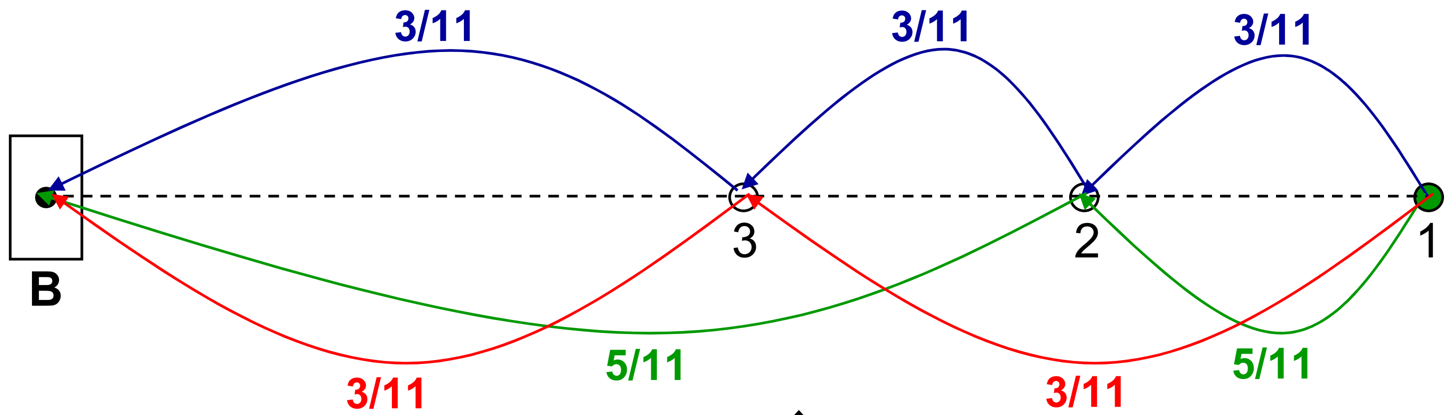
Solving the LP for large networks

- The number of route choices, |F|, is exponential in number of nodes, N
  - LP becomes too complex for large N
- Map to a network flow problem which can be solved in polynomial time
  - Allows finding the optimal route choices for a given network in polynomial time

Equivalence to Flow Problem

Route Choices: number of routes is exponential in N

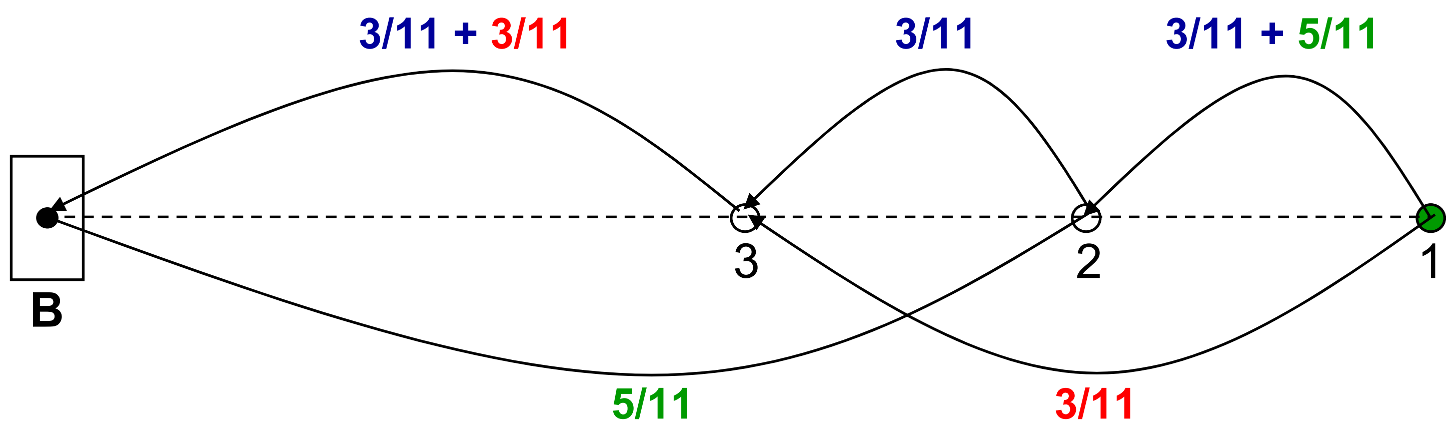
R_0: 0 (0)
R_1: 0.375 (3/11)
R_2: 0.375 (3/11)
R_3: 0.625 (5/11)
1.375 (11/11)



$\Updownarrow$

Network Flow View: number of possible edges is polynomial in N

$f_{1\rightarrow 2}$: 8/11
$f_{1\rightarrow 3}$: 3/11
$f_{1\rightarrow B}$: 0
$f_{2\rightarrow 3}$: 3/11
$f_{2\rightarrow B}$: 5/11
$f_{3\rightarrow B}$: 6/11



Slide adapted from: M Bhardwaj, INFOCOM 2002

Polynomial time LP

Objective :

$$\max \quad t$$

Constraints :

Non-negativity of flow:

$$f_{ij} \geq 0$$

Conservation of flow:

$$\sum_{\substack{s \in [1, N+1] \\ s \neq i}} f_{si} = \sum_{\substack{d \in [1, N+1] \\ d \neq i}} f_{id} : \quad i \in [2, N]$$

Total sensor flow:

$$\sum_{d \in [2, N+1]} f_{1d} - \sum_{s \in [2, N+1]} f_{s1} = 1$$

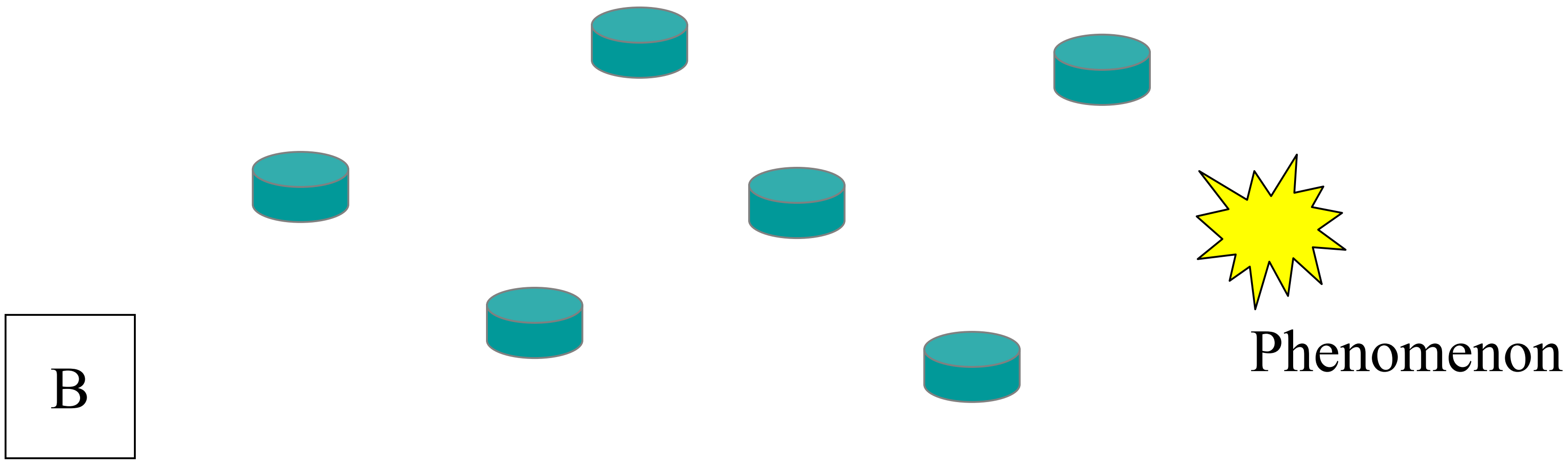
Energy constraints:

$$t \left(\sum_{\substack{d \in [1, N+1] \\ d \neq i}} p_{tx}(i, d) f_{id} + \sum_{\substack{s \in [1, N+1] \\ s \neq i}} p_{rx} f_{si} + \underbrace{p_{sense}}_{\textit{For node 1 only}} \right) \leq e_i : \quad i \in [1, N]$$

Sensing Lifetime

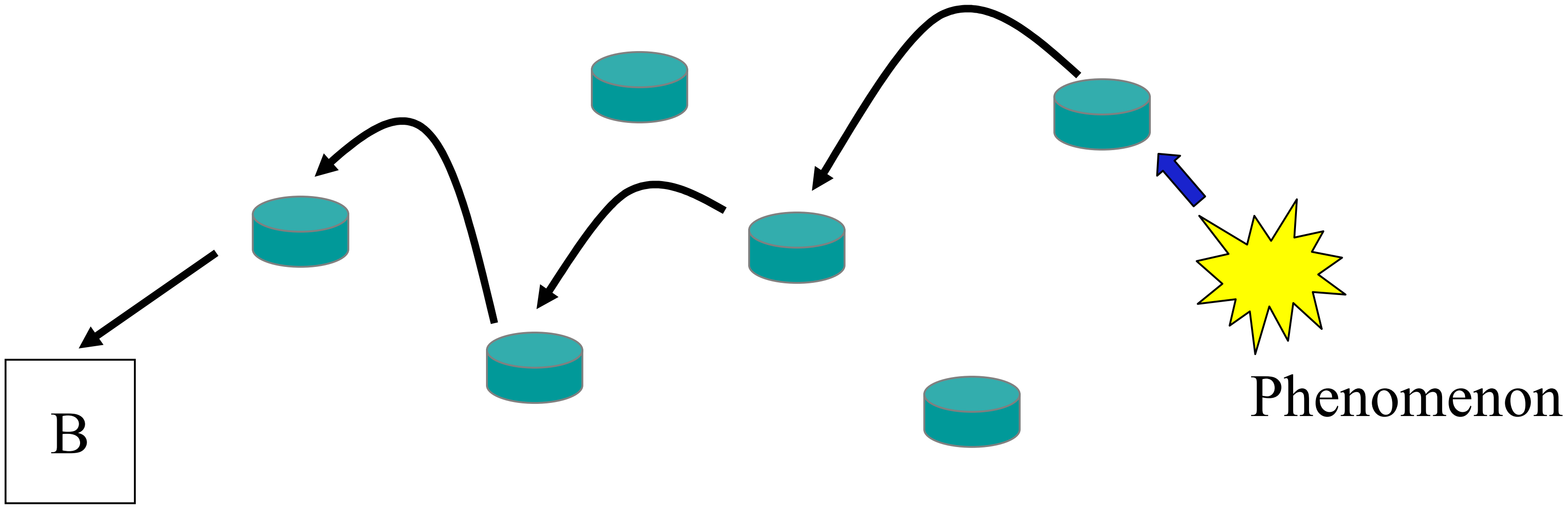
[Kansal et al, 2005]

- If data sources not specified, but given
  - Phenomenon to be sensed
  - Distortion quality required
- Choose the relevant sensors and determine the routes for maximum lifetime



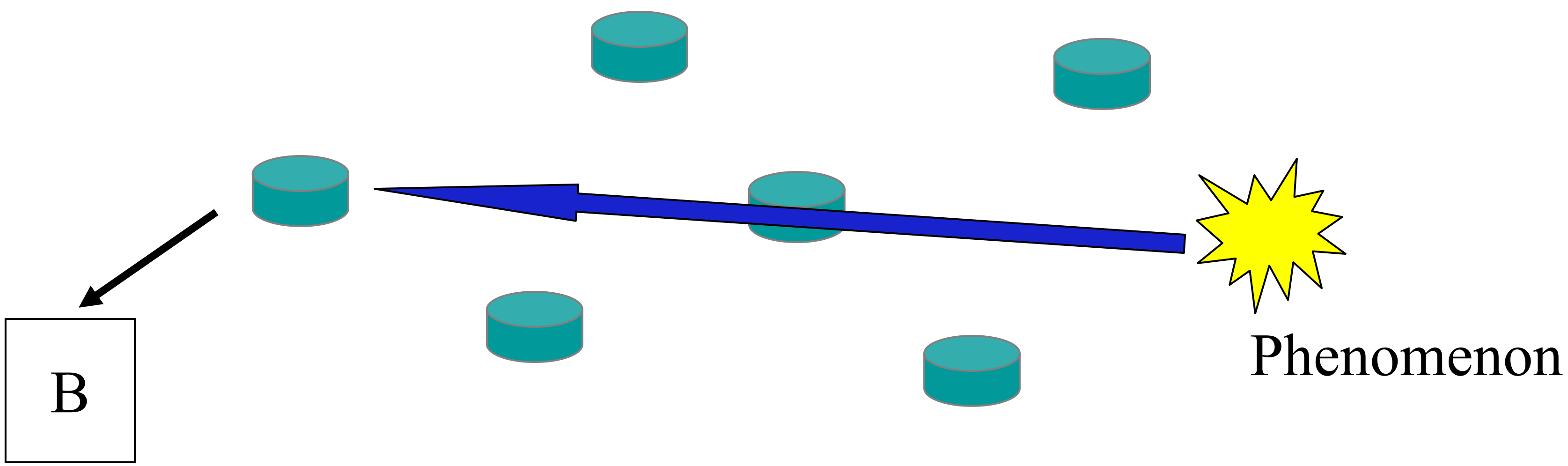
Choose closest sensor

- Small data size (since measurement quality is high)
- Large routing energy cost (possibly long route)



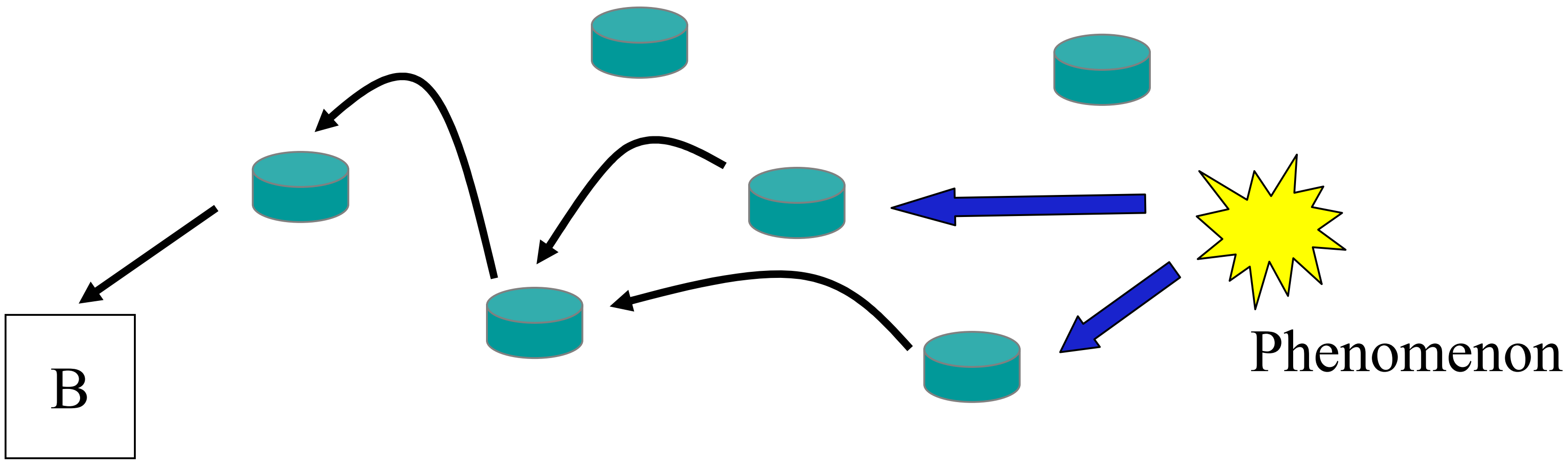
Choose sensor with smallest route

- Small routing energy cost
- Large data size (measurement quality is bad)



Collaborative sensing

- Use multiple sensors and fuse

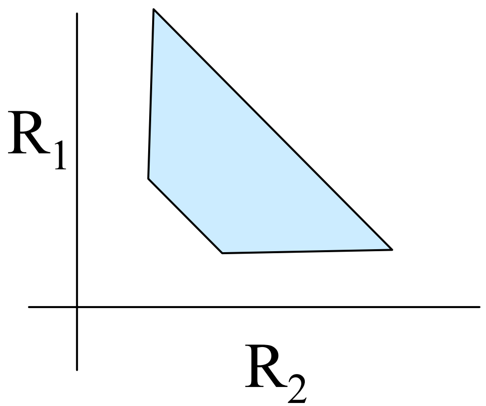


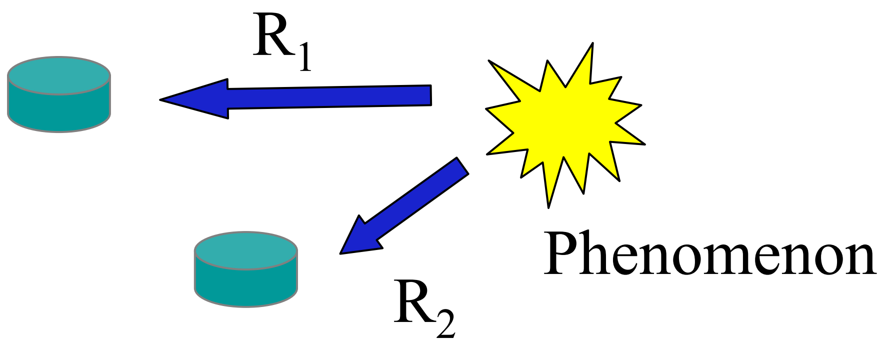
Lifetime-Distortion Relationship

- How to determine the best choice?
  - Combine lifetime maximization tools with data fusion analysis
- Find the possible choices of sensors for required distortion performance
  - Using data fusion analysis
- For each choice, find maximum achievable lifetime
  - using network flow LP
- Choose the sensors and routes which maximize lifetime

Sensor Selection

- All sensors within sensing range of the phenomenon are potential choices
  - How much data generated at each sensor: possible choices for given distortion are given by the rate allocation region of the Gaussian CEO result





$$\sum_{k \in A} R_k \geq \sum_{k \in A} r_k + \frac{1}{2} \log_2 \frac{1}{D} - \frac{1}{2} \log_2 \left[\frac{1}{\sigma_X^2} + \sum_{k \in A^c} \frac{1 - 2^{-2r_k}}{\sigma_k^2} \right] \forall \text{ non-empty } A \subseteq \{1, ..., N\}$$

$$\frac{1}{\sigma_X^2} + \sum_{k=1}^{N} \frac{1 - 2^{-2r_k}}{\sigma_k^2} \geq \frac{1}{D}$$

Energy Constraints for Chosen Sensors

- Same as before, but written for the variable source sensor rates
  - Maximize t as before

$$f_{ij} \geq 0, \quad R_i \geq 0, \quad r_i \geq 0, \forall i,j \in \{1,...,N\}$$

Flow Conservation:

$$\sum_{d \in [1,N+1], d \neq i} f_{id} - \sum_{s \in [1,N+1], s \neq i} f_{si} = R_i, \quad i \in \{1,...,N\}$$

Energy Constraints:

$$t \left[\sum_{d \in [1,N+1], d \neq i} P_{tx}(i,d) f_{id} + \sum_{s \in [1,N+1], s \neq i} P_{rx}(s,i) f_{si} + P_{sense} R_i \right] \leq E_i, \quad i \in \{1,...,N\}$$

Conclusions

- Network design depends on multiple methods
  - Optimal methods sometimes yield distributed protocols
  - Provide insight into the performance of available distributed methods
- Saw some examples of analytical tools that may help network design
- Open problem: Convergence of separate solutions